



Universidad
Carlos III de Madrid

Departamento de Ingeniería de Sistemas y
Automática

PROYECTO FIN DE CARRERA

Diseño de placas controladoras para mini-robots de bajo coste

Autor: Itziar Lima Ramos

Director: Alberto Valero Gómez

Tutor: Juan González Gómez

Leganés, septiembre 2012

Título: Diseño de placas controladoras para mini-robots de bajo coste

Autor: Itziar Lima Ramos

Director: Alberto Valero Gómez

Tutor: Juan González Gómez

EL TRIBUNAL

Presidente: César Arismendi Gutiérrez

Vocal: Cristina Fernández Herrero

Secretario: Juan Carlos González Vítores

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 27 de Septiembre del 2012 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Leganés, septiembre 2012

Agradecimientos

A mis padres y mi familia, por su apoyo, su comprensión y su paciencia.

A Juan Pablo, mi novio y mi mejor amigo, por su cariño, su ayuda y su apoyo constante. Por animarme siempre en los malos momentos. Porque sin él no hubiese conseguido terminarlo a tiempo.

A Juan y Alberto, por su ayuda y su paciencia.

A Shihi, mi "jie jie", por su apoyo durante todos estos años y su amistad. Por estar siempre ahí, en los momentos buenos y en los no tan buenos.

A mis amigos de la Universidad, porque son geniales. Por todos estos años que hemos pasado juntos. Porque siempre están ahí y sé que siempre lo estarán aunque no estemos en la misma ciudad.

A mis amigos y compañeros de beca, por todo el apoyo y los ánimos que me han dado, especialmente durante este último mes tan difícil.

Resumen

El objetivo de este proyecto final de carrera es el diseño de una placa controladora de bajo coste, basada en un microcontrolador de ATMEL¹.

Para dicho diseño se han implementado dos versiones, añadiendo en cada una de ellas distintas funcionalidades. Así, posteriormente, dependiendo de la funcionalidad que se requiera se utilizará una placa u otra.

La primera versión, la más sencilla, es la **Skymega1.0**. Esta placa permite controlar un robot de forma simple. Utiliza un microcontrolador AtmegaX8, donde X es la capacidad de memoria disponible para programa: 4, 8, 16 o 32 KBytes; en concreto, se ha usado el Atmega168. Consta de un puerto serie, bus I2C, ocho conectores de tres pines: un pin digital acompañado de alimentación y GND, (pensados especialmente para servos), LED y pulsador para pruebas, y un puerto de expansión para su comunicación con otros dispositivos –con otras placas por ejemplo-.

La siguiente versión, **Skymega2.0**, ha sido mejorada respecto a la primera introduciendo la electrónica necesaria para poder conectar y controlar dos motores de continua. Además, se han añadido otros conectores, uno para Bluetooth y cinco cuyo objetivo inicial es usar tres para la conexión de sensores infrarrojos y dos para sensores de ultrasonidos; aunque estos conectores pueden ser usados para cualquier otro fin. Junto a estos conectores, se encuentran otros cuatro que inicialmente se han destinado a conectar servos, -pueden ser también usados para otro fin, como por ejemplo para conectar sensores infrarrojos-. En esta placa, al igual que en la 1.0, el microcontrolador será un AtmegaX8. En este caso se ha usado el Atmega328.

Paralelamente, se ha realizado un pequeño tutorial de cómo usar ciertas herramientas para el diseño de placas, con el fin de facilitar a aquellas personas que se inician en este campo la labor tanto de diseño como de rutado, siendo ésta última una tarea a veces complicada y lenta.

Palabras clave: robots, placa, Skymega1.0, controlar, Skymega2.0, microcontrolador, sensores, infrarrojo, ultrasonidos, conectores, motores, comunicación.

¹ **Atmel** es una compañía de semiconductores, fundada en 1984. Su línea de productos incluye microcontroladores (incluyendo derivados del 8051, el AT91SAM basados en ARM, y sus arquitecturas propias AVR y AVR32), dispositivos de radiofrecuencia, memorias EEPROM y Flash, ASICs, WiMAX, y muchas otras. Dentro de la familia AVR se encuentran el ATMEGA 168 y 328P que se usa en este proyecto.

Abstract

The aim of this project is to limit the design of an affordable controller board based on an Atmel microcontroller.

Two versions have been implemented including different functionalities in each one. Thus, depending on the required functionality we will choose a different board for the robot.

The first version, the simplest, is the Skymega1.0. This board allows to control a robot in a simple way. It uses a microprocessor AtmegaX8, where X is the memory capacity for program: 4, 8, 16 or 32 Kbytes. In this board the Atmega168 has been used. The board has a serial port, I2C bus, eight three-pin connectors: a digital pin, a power pin and a GND pin, (designed especially for servos), LED and pushbutton for testing, and an expansion port to communicate other devices - For instance, with other boards.

The second version, Skymega2.0 has been improved adding the electronics needed to connect and control two DC motors. Additionally, other connectors have been included, one for Bluetooth, three to connect infrared sensors and two to connect ultrasonic sensors, however these last five connectors could be used for other purposes. The board includes also four extra connectors which were initially thought to connect servos, but they can also be used for other purposes, such as connecting infrared sensors. This second board has an Atmega328 microcontroller.

Finally, a small tutorial has been created to control how to use certain tools for the design of boards, facilitating the boards development tasks.

Keywords: robots, board, Skymega1.0, to control, Skymega2.0, microcontroller, sensors, infrared, ultrasound, connectors, motors, communication.

Índice general

Capítulo 1	Introducción y objetivos.....	17
1.1	Introducción	18
1.2	Objetivos	18
1.3	Fases del desarrollo	18
1.4	Medios empleados.....	20
1.5	Estructura de la memoria	20
Capítulo 2	Soluciones existentes.....	22
2.1	Introducción	23
2.2	Soluciones analizadas	23
2.2.1.	<i>Placas Shield</i>	23
2.2.2.	<i>Placas controladoras</i>	26
2.3	Conclusiones.....	29
Capítulo 3	Diseño de las placas	32
3.1	Skymega1.0.....	33
3.1.1.	<i>Introducción</i>	33
3.1.2.	<i>Uso</i>	33
3.1.3.	<i>Descripción y Componentes</i>	34
3.1.4.	<i>Diseño de la placa</i>	36
3.1.5.	<i>Alimentación de la Skymega1.0</i>	44
3.1.6.	<i>Carga del bootloader</i>	46
3.1.7.	<i>Conexión con el PC</i>	47
3.1.8.	<i>Conexiones de la placa</i>	49

3.1.9. Pruebas.....	51
3.1.10. Resultado final	53
3.1.11. Mejoras	55
3.2 Skymega2.0.....	56
3.2.1. Introducción.....	56
3.2.2. Uso	56
3.2.3. Descripción y Componentes.....	56
3.2.4. Diseño de la placa	59
3.2.5. Alimentación de la Skymega2.0	72
3.2.6. Carga del bootloader.....	73
3.2.7. Conexión con el PC	73
3.2.8. Conexiones de la placa	74
3.2.9. Pruebas.....	75
3.2.10. Resultado final	76
3.3 Comparativa entre las dos placas.....	76
3.3.1. Comparativa de funcionalidades	77
3.4 Trabajos futuros.....	78
3.4.1. Mejora de la carga del bootloader y del código en el microprocesador.	78
3.4.2. Conector de un módulo XBee	80
3.4.3. Posibilidad de conexión de una LiPo de más celdas	80
Capítulo 4 Manual KiCad.....	83
4.1 Introducción	84
4.2 Diseño del esquemático	84
4.3 Generación de la Netlist	86
4.4 Circuito impreso	89

4.5 Rutado	92
4.6 Imprimir las capas de la placa	98
4.7 Generar los archivos Gerber	98
Capítulo 5 Estudio de costes	101
5.1 Introducción	102
5.2 Comparativa general entre placas Skymega	102
5.3 Precio componentes programación	103
3.1. Referencias Skymega1.0	118
3.2. Referencias Skymega2.0	119
4.1. TC54	121
4.2. Módulo PTH08000WAZT	121
4.3. Integrado L293	121
4.4. Microcontrolador Atmega328/168	121

Índice de figuras

Imagen 1. Motor Control V1.1	23
Imagen 2. Arduino Ethernet Shield	24
Imagen 3. Arduino Relay Shield	25
Imagen 4. Arduino grblShield v3	25
Imagen 5. Rugged Motor Drive Shield	26
Imagen 6. Placa-robot DFRobot Rover V2.....	26
Imagen 7. BotBoarduino	27
Imagen 8. Arduclema	27
Imagen 9. BlimpDuino	28
Imagen 10. Romeo	29
Imagen 11. Miniskybot con la placa Skymega1.0.....	33
Imagen 12. Módulo Repyl con la placa Skymega1.0	34
Imagen 13. Esquemático Skymega1.0	36
Imagen 14. Detalle esquemático del Atmega168.....	37
Imagen 15. Detalle esquemático de la alimentación de la placa.....	38
Imagen 16. Detalle esquemático del LED de encendido	38
Imagen 17. Detalle esquemático del filtro de ruido	38
Imagen 18. Detalle esquemático del puerto ICSP.....	39
Imagen 19. Detalle esquemático del puerto serie.....	39
Imagen 20. Detalle esquemático de los conectores de los servos	40
Imagen 21. Detalle esquemático del puerto de expansión	41
Imagen 22. Detalle esquemático del LED de prueba.....	41
Imagen 23. Detalle esquemático del pulsador de prueba	42
Imagen 24. Detalle esquemático del bus I2C.....	42
Imagen 25. Skymega1.0 rutada.....	43
Imagen 26. Archivos fabricación Skymega1.0	43
Imagen 27. Cable de alimentación Molex.....	44
Imagen 28. Portapilas con el conector Molex.....	44
Imagen 29. Conexión del Cable Molex en la Skymega1.0	45
Imagen 30. Alimentación Skymega1.0 con portapilas imprimible	45
Imagen 31. Cable USB- Molex	46
Imagen 32. Alimentación de la placa Skymega1.0 con fuente de alimentación.....	46
Imagen 33. Conexión de la placa Skymega1.0 con el programador AVR.	47
Imagen 34. Detalle de la conexión de la placa Skymega1.0 conector ICSP	47
Imagen 35. Detalle Cable USB-serie FTDI.....	48
Imagen 36. Conexión Skymega1.0 cable USB- serie FTDI	48
Imagen 37. Cable USB-serie FTDI	49
Imagen 38. Conector del servo Futaba 3003.....	49
Imagen 39. Servo Futaba 3003 conectado a la placa Skymega1.0	50
Imagen 40. Skymega1.0 y FreeLEDs conectadas mediante el puerto de expansión	51
Imagen 41. Skymega1.0 y Skypads conectadas mediante el puerto de expansión	51
Imagen 42. Primer prototipo Skymega1.0	52
Imagen 43. Segundo prototipo de la Skymega1.0.	52
Imagen 44. Tercer prototipo Skymega1.0	53
Imagen 45. Placa Skymega 1.0.....	54
Imagen 46. Esquemático Skymega2.0	60
Imagen 47. Detalle esquemático del Atmega328.....	60

Imagen 48. Detalle esquemático de la alimentación de la placa.....	61
Imagen 49. Detalle esquemático del filtro de ruido	62
Imagen 50. Detalle esquemático de los conectores de los servos	63
Imagen 51. Detalle esquemático del puerto de expansión	64
Imagen 52. Detalle esquemático de los conectores de los sensores ultrasonido.....	64
Imagen 53. Detalle esquemático de los conectores de los sensores infrarrojo	65
Imagen 54. Detalle esquemático del interruptor Switch PC4	65
Imagen 55. Detalle esquemático del interruptor Switch PC5	66
Imagen 56. Detalle esquemático del LED de prueba.....	66
Imagen 57. Detalle esquemático del pulsador de prueba	67
Imagen 58. Detalle esquemático del puerto ICSP.....	67
Imagen 59. Detalle esquemático del puerto serie.....	68
Imagen 60. Detalle esquemático del bus I2C.....	68
Imagen 61. Detalle esquemático del conector Bluetooth	69
Imagen 62. Detalle esquemático control del motor	69
Imagen 63. Colocación componentes en la placa Skymega2.0	70
Imagen 64. Placa Skymega2.0 rutado en Specetra.....	71
Imagen 65. Placa Skymega2.0 ya rutada e importada a KiCad	71
Imagen 66. Archivos de fabricación placa Skymega2.0	72
Imagen 67. Conector Jack	72
Imagen 68. Batería LiPo de 7,4V (dos celdas).....	73
Imagen 69. Módulo FTDI	74
Imagen 70. Placa prototipo Skymega2.0.....	76
Imagen 71. Detalle esquemático del circuito USB	78
Imagen 72. Prueba hardware circuito USB	80

Índice de tablas

Tabla 1. Componentes de la Skymega1.0.....	35
Tabla 2. Puntos débiles Skymega1.0	55
Tabla 3. Componentes de la placa Skymega2.0.....	57
Tabla 4. Comparativa de funcionalidades entre Skymega1.0 y Skymega2.0	77
Tabla 5. Componentes del circuito USB	79
Tabla 6. Comparación de costes de material de programación	79
Tabla 7. Comparación de precios entre Skymega1.0 y Skymega2.0	102
Tabla 8. Materiales de programación Skymega1.0 y 2.0	103

Capítulo 1

Introducción y objetivos

1.1 Introducción

Como ya se ha indicado en el resumen, el proyecto ha consistido en el diseño de dos placas, cuyo principal fin será su uso en mini-robots de bajo coste, aunque la capacidad de éstas no está reducida a este fin, pudiendo tener cualquier otro uso dentro de sus capacidades.

El punto de partida ha sido la placa Skymega1.0. Tras hacer pruebas y conocer las necesidades de la gente, ha sido mejorada creándose así la Skymega2.0.

De esta forma según el uso que se le vaya a dar a la placa, o en concreto, el propósito del robot, se podrá adquirir una u otra, siendo la última versión la más completa, aunque también más cara.

Más adelante se procederá a describir con detalle cada una de ellas.

1.2 Objetivos

El objetivo fundamental del proyecto es diseñar una placa de bajo coste para el control de mini-robots de bajo coste. El uso de estas placas no queda reducido únicamente a su uso en robótica como ya se ha indicado.

En base a ese objetivo principal, se proponen los siguientes objetivos parciales:

- Diseñar y fabricar la placa Skymega1.0
- Diseñar y fabricar la placa Skymega2.0

Como segundo objetivo, se plantea la tarea de realizar un pequeño tutorial de las herramientas de diseño utilizadas, de tal manera que se facilite la labor de diseño y rutado a aquellas personas que se inician en este campo.

1.3 Fases del desarrollo

Las diferentes fases del desarrollo han sido:

- **Identificación de las necesidades**

En primer lugar se han identificado las necesidades hardware de un robot móvil de bajo coste en términos de procesamiento, entradas y salidas.

- **Estudio del mercado**

Se ha realizado un estudio de las soluciones existentes en el mercado para comprobar si estas necesidades están ya cubiertas².

² En el Capítulo 2 se realizará un breve análisis del mercado, comprobando qué necesidades aún no están cubiertas.

- **Diseño de la placa en papel**

Una vez decidido lo que se quiere hacer, se realizará un diseño preliminar en papel, con el objetivo de concretar los objetivos y necesidades.

- **Elección de los componentes**

Cuando ya está el primer diseño hecho, se elegirán los componentes que se van a utilizar, mirando las hojas de características y su coste.

- **Montaje del diseño en una placa protoboard**

Una vez se tienen todos los componentes, se realizará el montaje sobre una placa protoboard, para comprobar si el diseño es correcto y la placa tiene la funcionalidad esperada.

- **Pruebas hardware y software de la placa**

Se realizarán una serie de pruebas hardware y software de la placa para comprobar la funcionalidad y su correcto funcionamiento.

- **Diseño del esquemático de la placa en KiCad**

Una vez comprobado que el diseño de la placa es el correcto y funciona, se realizará el diseño del esquemático en KiCad.

- **Rutado de la placa**

Una vez se ha realizado el esquemático se rutará la placa como se indica en el manual de KiCad, en este mismo documento.

- **Fabricación de la placa**

Una vez ya está la placa rutada y se tienen los archivos de fabricación, se mandará a fabricar la placa. Inicialmente se fabricará un prototipo para después fabricar una tira de placas a nivel industrial.

- **Pruebas hardware y software de la placa definitiva**

Con la placa ya fabricada sobre una PCB, se realizarán una serie de pruebas hardware y software para comprobar que funciona correctamente.

1.4 Medios empleados

Medios hardware empleados en la realización del proyecto

Los medios hardware con los que se ha contado para realizar el proyecto han sido:

- Componentes electrónicos y cable
- Placa protoboard
- Placas de puntos
- Cable USB
- Polímetro
- Ordenador

Medios software empleados en la realización del proyecto

Los medios software con los que se ha contado para realizar el proyecto han sido:

- Programa de diseño de PCB KiCad
- Linux

1.5 Estructura de la memoria

En el **Capítulo 1** se realiza una introducción del proyecto indicando además los objetivos, las diferentes fases del desarrollo, los medios empleados para la realización del proyecto y la estructura de la memoria.

En el **Capítulo 2** se analizarán las soluciones ya existentes en el mercado. De esta forma se buscarán las necesidades aun no cubiertas, y en función de esto, se tomarán las decisiones para el diseño de las placas.

En el **Capítulo 3** se describe con detalle el diseño de las placas, indicando de cada una de ellas el uso que va a tener, la descripción de los componentes y el esquemático de la placa -explicando la función de cada bloque-.

Se explicará además con detalle la carga del bootloader, la alimentación y las conexiones de las placas. Adicionalmente, se describirán las pruebas software y hardware realizadas, así como las mejoras propuestas.

En el **Capítulo 4** se recoge un manual del programa de diseño de placas electrónicas, el software libre KiCad, mediante el cual se puede realizar una placa de principio a fin, desde el esquemático de la placa hasta obtener los archivos de fabricación de ésta.

En el **Capítulo 5** se hará una comparativa de costes entre las dos placas diseñadas.

Capítulo 2

Soluciones existentes

2.1 Introducción

Antes de comenzar el desarrollo y diseño de las placas, se ha realizado un estudio de las soluciones ya existentes, de tal forma que se pueda justificar el diseño de una nueva placa que cubra aquellas necesidades aun no contempladas.

Debido a la gran variedad de placas, el criterio de búsqueda ha sido analizar aquellas soluciones que por tamaño y entorno de desarrollo –Arduino-, se encuentren dentro de nuestras expectativas.

Se ha buscado hardware libre, compatible con el entorno Arduino, que permita la conexión de diferentes sensores, posibilidad de comunicación con otros dispositivos, control de motores de continua, bajo coste, flexibilidad en la alimentación y pequeño/mediano tamaño.

2.2 Soluciones analizadas

Según los criterios expuestos, y teniendo en cuenta la amplitud del mercado, las soluciones analizadas han sido las siguientes:

2.2.1. Placas Shield

2.2.1.1. Motor Control V1.1

Esta placa es una shield pensada para ser conectada sobre la placa Arduino Uno que permite el control de hasta cuatro motores de continua –sentido de giro y velocidad- por medio del integrado L293.

Esta shield no dispone de etapa de potencia propia, aprovechándose de la etapa de potencia disponible en la placa de Arduino.

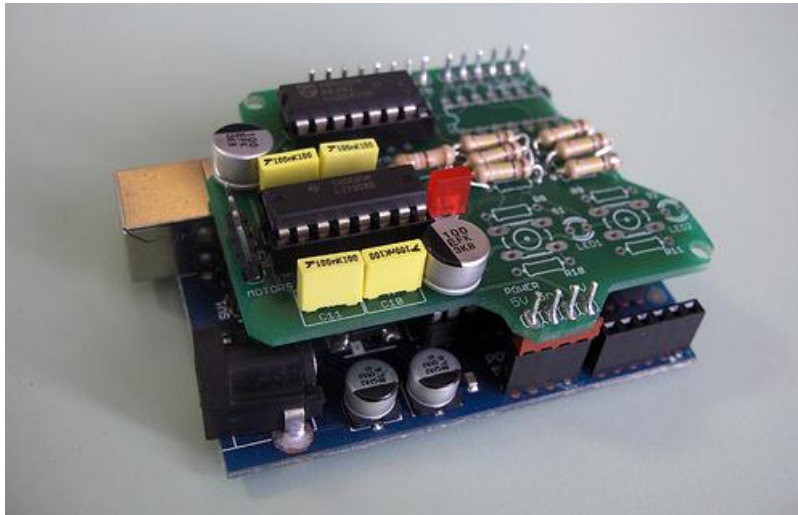


Imagen 1. Motor Control V1.1

2.2.1.1. Arduino Ethernet Shield

La Arduino Ethernet Shield permite a una placa Arduino conectarse a internet. Se trata de una shield pensada para ser conectada sobre una placa Arduino Uno. Está basada en el chip Ethernet Wiznet W5100 que soporta hasta cuatro conexiones de sockets simultáneas.



Imagen 2. Arduino Ethernet Shield

2.2.1.1. Arduino Relay Shield

Esta placa permite controlar, mediante una placa Arduino, cuatro relés para conmutar cargas externas tales como bombillas, motores etc. Incluye un zócalo para un módulo XBee que permite establecer una comunicación inalámbrica para controlar la placa.

No tiene etapa de potencia, y se recomienda el uso de una fuente externa que aporte mayor corriente y la tensión adecuada.



Imagen 3. Arduino Relay Shield

2.2.1.1. Arduino grblShield v3

Esta placa permite el control de tres motores paso a paso que requieran una gran potencia y ocho motores paso a paso de menor potencia. Se trata de una shield para la placa Arduino Uno o la Duemilanove. La alimentación debe realizarse externamente, soportando una corriente de hasta 2.5A y un rango de tensiones de entrada de 12 a 30V.

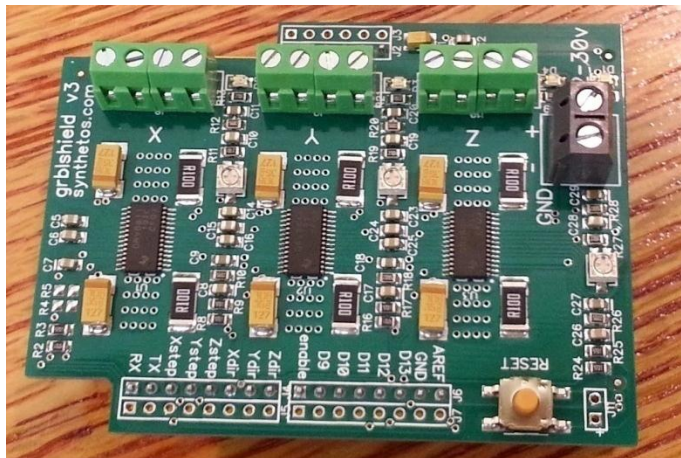


Imagen 4. Arduino grblShield v3

2.2.1.2. Rugged Motor Driver Shield

Esta placa está diseñada para el control de dos motores de continua o un motor bipolar paso a paso. La tensión de entrada deberá ser de 30V y soporta una corriente de hasta 2,8A.

Es compatible con las placas Arduino Uno, Mega y Duemilanove, aunque también puede ser usada de manera independiente.

Leva protección contra tensión inversa, sobre-corriente y sobre-temperatura.

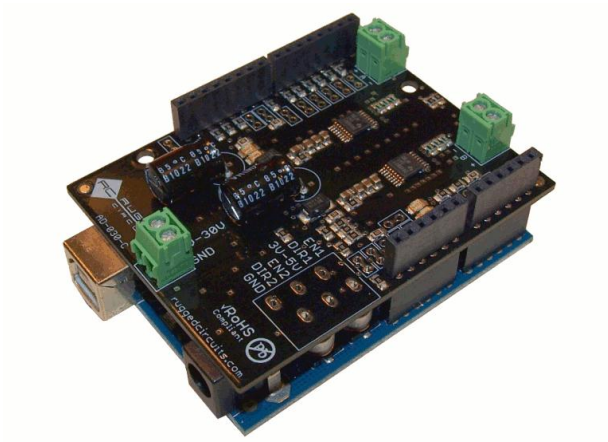


Imagen 5. Rugged Motor Drive Shield

2.2.2. Placas controladoras

2.2.2.1. DFRobotShop Rover V2

En este caso no se trata de una simple placa. Es una placa-robot basada en la placa Arduino Uno. Dispone del ATmega328, el integrado L293 para el control de motores de continua, regulador de tensión, cargador de batería LiPo y la posibilidad de alimentarla entre 3,7 y 9V.



Imagen 6. Placa-robot DFRobot Rover V2

2.2.2.2. BotBoarduino

BotBoarduino es una placa compatible con Arduino, hecha específicamente para los robots Lynxmotion. Conserva la posición de los pines de Arduino.

Puede alimentarse externamente a través del USB o de un conector específico, pensado especialmente para suministrar la corriente necesaria para varios servos.



Imagen 7. BotBoarduino

2.2.2.3. Arduclema

Arduclema es una placa basada en Arduino Diecimila. Es totalmente compatible con Arduino Diecimila y se diferencia, básicamente, en que Arduclema incorpora unas clemas de tornillo para unas conexiones eléctricas más robustas y fiables.

Otra característica del Arduclema es que permite comunicación I2C.

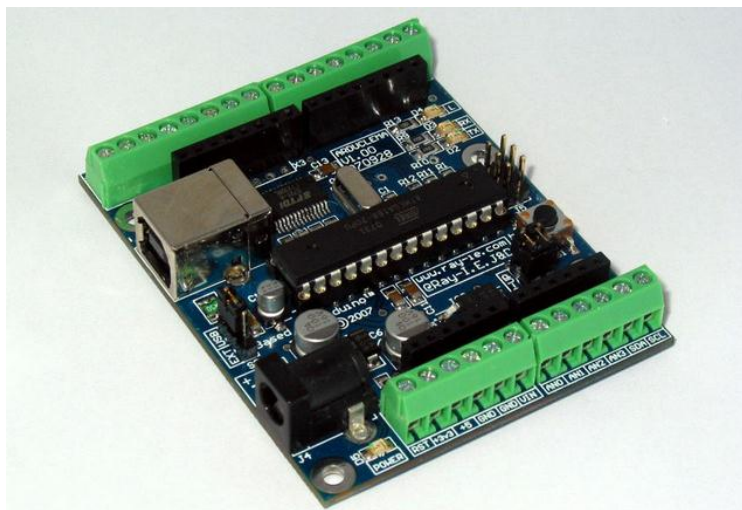


Imagen 8. Arduclema

2.2.2.4. BlimpDuino

Es una pequeña placa, dotada con tres sensores IR integrados, un conector para sensor ultrasonidos, conector para un servo y un integrado para el control de un dos motores de continua. Está diseñada para ser alimentada mediante una pila

LiPo de dos celdas (7,4V). Para el control de descarga de ésta, lleva un detector de low voltage que evita descargas profundas.

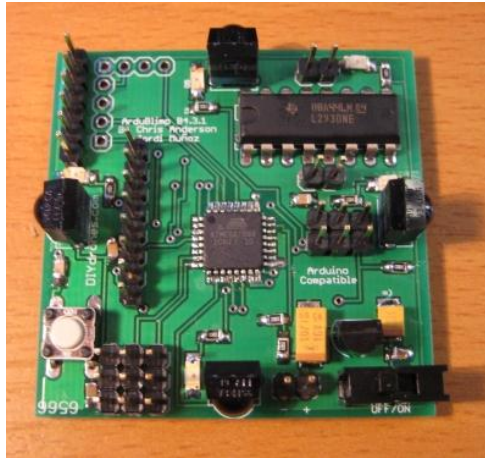


Imagen 9. Blimpduino

2.2.2.5. ROMEO

La placa Romeo es un microcontrolador todo-en-uno diseñado especialmente para aplicaciones de robótica. Posee los beneficios de la plataforma Arduino, miles de códigos fuente de ejemplo con licencias libres. Además se puede ampliar fácilmente con la mayoría de los Escudos Arduino. Por otro lado, integra control para dos motores DC y puerto para conexión inalámbrica que proporciona una forma mucho más fácil para comenzar su proyecto de robótica.

Lo más importante de esta placa es que es compatible con Arduino lo que significa que puede utilizar Romeo como una Arduino Nano standard.

Algunas especificaciones de esta placa son las siguientes:

- Atmega328
- Catorce E/S digitales
- Seis canales PWM Canales (Pin11 , PIN10, pin9, Pin6, Pin5, pin 3)
- Ocho canales E / S analógicas de 10 bits
- Interfaz USB
- Auto detección / conmutación de entrada de potencia
- Cabecera ICSP para programación directa del microcontrolador
- La interfaz en serie TTL
- Conexiones tanto pin macho como pin hembra
- Conexiones integradas para módulo APC220 RF y módulo Bluetooth DF-
- Cinco puertos I2C
- Control de dos motores DC máx 2Ad
- Siete botones
- Alimentación DC: con alimentación USB o externo 7V ~ 12V DC

- Salida de CC: 5 V / 3,3 V DC y la salida de alimentación externa
- Dimensiones: 90x80mm
- Peso: 60 gramos

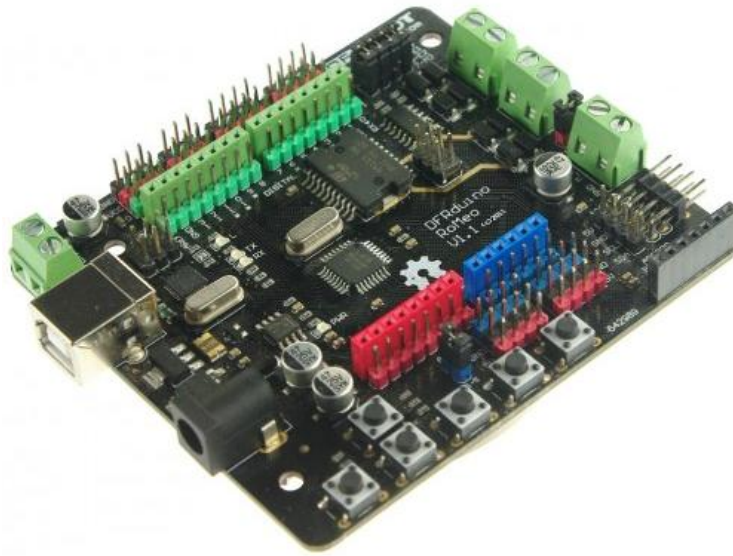


Imagen 10. Romeo

2.3 Conclusiones

Después de analizar estas diez placas, se puede observar que la variedad y las posibilidades disponibles son elevadas.

Se podría decir que para cualquier necesidad existe una solución, pero, ¿existe una placa que cubra el mayor número de necesidades en una sola placa?

Para responder a la pregunta, se analizan las placas anteriormente descritas y se observa que cada una de ellas cubre un aspecto o necesidad: control de motores, comunicación Ethernet, flexibilidad en las conexiones, etcétera; pero casi ninguna cubre varias necesidades a la vez.

La mayoría de ellas son shield's que dependen de otra placa Arduino, bien sea Uno, Duemilanove o Mega, y pocas ofrecen una gran funcionalidad por ellas mismas.

Por otro lado, la placa Romeo es muy versátil. Tiene un gran número de funcionalidades, y de pines duplicados. Por ello la placa tiene unas dimensiones grandes (90x80mm), por lo que no permite colocarla en un robot pequeño. Además, no se pueden conectar a ella directamente sensores, o servos, ya que no tienen conectores preparados para ello.

Por lo tanto, a pesar del amplio abanico disponible, no existe una placa que cubra por si sola todos estos aspectos sin necesidad de estar conectada a otras.

El objetivo de este proyecto será diseñar una placa que contenga, en la medida de lo posible, todas las anteriores en una única placa, y a su vez, que sea de bajo coste.

Aunque en el análisis anterior no se ha hecho mención al precio de las placas, su coste ronda entre los 20 € –para las placas con menos funcionalidad- y los 60 € - para aquellas placas más complejas-.

Nuestro objetivo será conseguir que el coste de la placa más completa (la Skymega2.0) sea del orden de los 30 €, precio muy inferior al de las placas más complejas analizadas y a la vez menos completas que la que se diseñará en este proyecto.

Capítulo 3

Diseño de las placas

3.1 Skymega1.0

3.1.1. Introducción

Se trata de una tarjeta microcontroladora de reducidas dimensiones 514 x 514mm. La tarjeta Skymega1.0 es hardware libre y ha sido diseñada con la herramienta libre KiCad. También es compatible con Arduino.

3.1.2. Uso

Las aplicaciones principales son la programación de robots modulares y pequeños robots móviles, como el Miniskybot³ (o derivados), o bien para usos educativos.

A continuación se puede observar una imagen del robot imprimible Miniskybot.

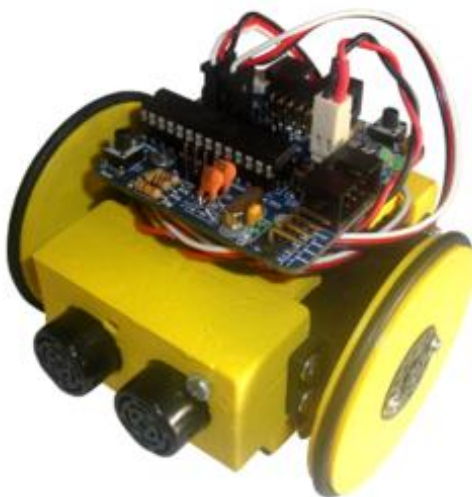


Imagen 11. Miniskybot con la placa Skymega1.0

En la siguiente imagen se puede observar el módulo Repy1⁴ con la placa Skymega1.0.

³ El robot Miniskybot ha sido realizado por Juan González Gómez. Se trata de un robot imprimible, fabricado mediante la impresora 3D.

⁴ El módulo Repy1 se trata de un módulo perteneciente a un robot serpiente modular que ha sido realizado por Juan González Gómez.



Imagen 12. Módulo Repy1 con la placa Skymega1.0

3.1.3. Descripción y Componentes

3.1.3.1. Componentes de la placa

En la tabla mostrada a continuación se encuentran los componentes necesarios para la fabricación de la Skymega1.0.

Componente	Valor	Cantidad	Referencia
Condensador cerámico/multicapa	22pF	2	C1, C2
Condensador cerámico/multicapa	100nF	2	C4, C5
Condensador electrolítico	1uF/16v	1	C3
Conector acodado 2x5 pines (IDC)	-	1	CT1
Conector acodado 2x3 pines (IDC)	-	1	H3
LED de 3mm	Verde	1	D1
LED de 3mm	Rojo	1	D2
Conector molex macho recto de 2 pines	2,54mm	1	M1
Conector molex macho acodado de 2 pines	2,54mm	1	M2
Tira de 4 pines macho rectos (1x4)	-	2	M4,M5

Tira de 4 pines macho acodados (1x4)	-	1	M6
Tira de 3 pines macho rectos	-	8	Servos
Resistencia de 1/4W	10K	1	R1
Resistencia de 1/4W	680	2	R3, R4
Resistencia de 1/4W	4K7	2	R5, R6
Pulsador para circuito impreso	-	2	S1, S2
Interruptor de 2 posiciones acodado para CI	-	1	SW1
Cristal de cuarzo	16MHz	1	X1
Zócalo de 2x14 pines (300 mils)	-	1	U1
Microcontrolador Atmega 168 de ATMEL	-	1	U1

Tabla 1. Componentes de la Skymega1.0

3.1.3.2. Conexiones con el microprocesador.

Los pines disponibles son del PB0 al PB5, del PC0 al PC5 y del PD0 al PD7. A ellos irá conectado lo siguiente:

- Pulsador de prueba
- LED de prueba
- Conector Bus I2C
- Puerto de Expansión
- Conectores 8 servos
- Conector ICSP
- Conector Puerto Serie

El *pulsador* y el *LED de prueba* se usarán, como su propio nombre indica, para realizar pruebas a la hora de programar. Se realizarán programas sencillos para comprobar el correcto funcionamiento de la placa.

El LED de “power on” indicará si la placa está encendida o apagada. Ésta se podrá encender o apagar mediante el interruptor de dos posiciones acodado.

El *Bus I2C* es un bus de comunicaciones serie. Por el pin 2, SDA, se transmitirán datos y por el pin 3 SCL, la señal de reloj.

El *Puerto de Expansión* se utiliza para conectar otras placas a la Skymega1.0.

Esta placa tiene ocho conectores. Cada uno de ellos está formado por un pin I/O (entrada/salida digital; entrada analógica), un pin de alimentación -VCC- y un pin de GND. Estos conectores podrán ser usados para la conexión de servos, sensores IR o ultrasonidos.

El puerto ICSP sirve para cargar el bootloader al microprocesador. Esto se realizará una única vez, cuando el microcontrolador es virgen. Si ya se ha utilizado previamente no hay que volver a cargarlo.

A través del Puerto Serie se le cargarán al microprocesador los programas.

3.1.4. Diseño de la placa

El diseño de la placa se ha realizado mediante el programa libre KiCad.

Se ha realizado un esquemático y posteriormente se ha rutado. A continuación se explicarán los pasos para la realización del esquemático y del rutado de la placa.

3.1.4.1. Esquemático

A continuación se muestra el esquemático de la placa Skymega1.0 realizado con KiCad.

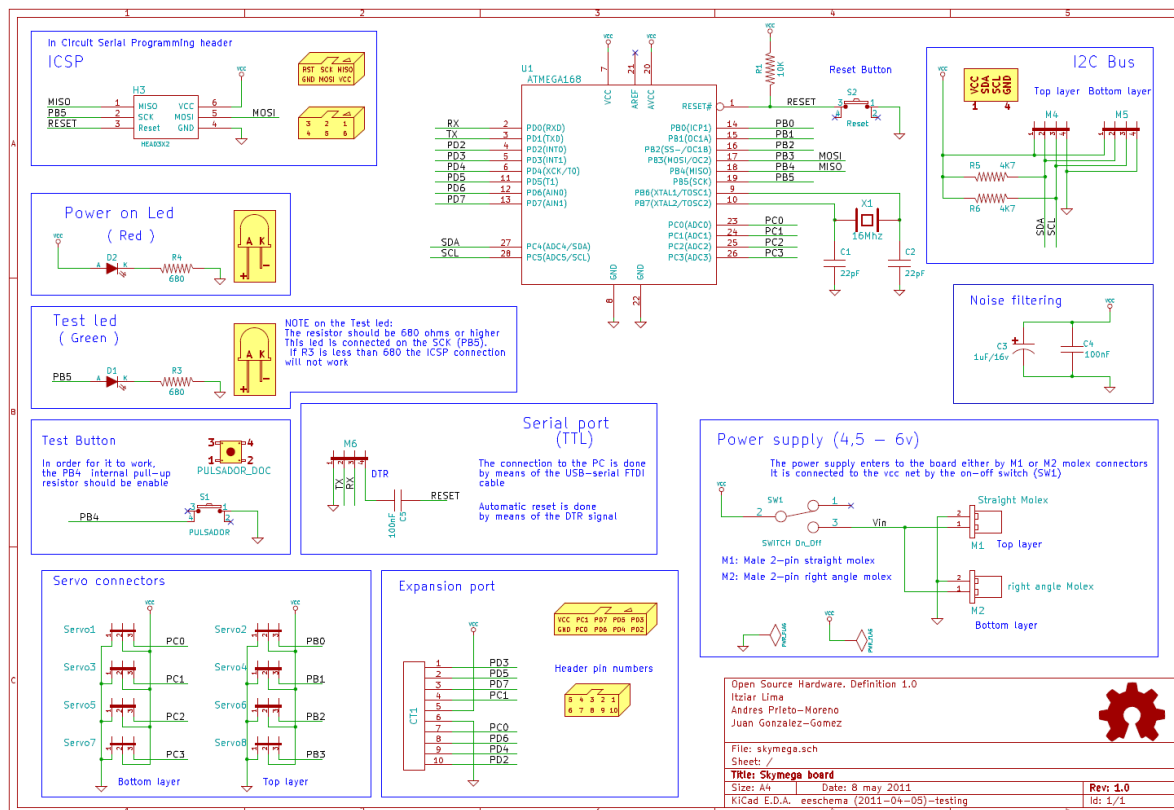


Imagen 13. Esquemático Skymega1.0

3.1.4.2. Criterios de diseño

Seguidamente se explicará los componentes y el criterio elegido.

Microprocesador

El microprocesador utilizado será el Atmega 168. En la imagen siguiente se pueden observar las conexiones de éste.

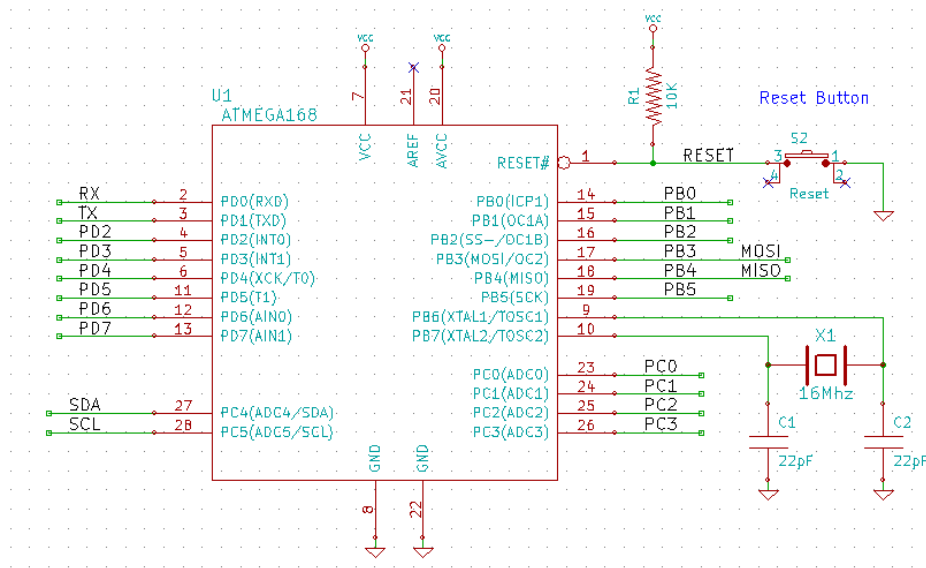


Imagen 14. Detalle esquemático del Atmega168

El cristal de 16MHz se conectará a los pines 9 y 10.

El pin 1 es el pin de Reset, al que se le conectará un pulsador mediante el cual se podrá resetear el microprocesador.

El puerto B (PB7... 0) es un puerto I/O de 8 bits bidireccional.

El puerto C (PC5... 0) es un puerto I/O de 6 bits bidireccional.

El puerto D (PD7...0) es un puerto I/O de 8 bits bidireccional.

Más adelante se describirán las conexiones con estos puertos.

Alimentación

La tensión de alimentación deberá encontrarse entre 4,5V y 6V. Se realizará a través de un conector Molex macho que se encuentra en la placa.

Se ha colocado un interruptor que permite encender y apagar la placa sin necesidad de desconectar la alimentación.

Más adelante se especificará cómo debe realizarse la alimentación de la placa.

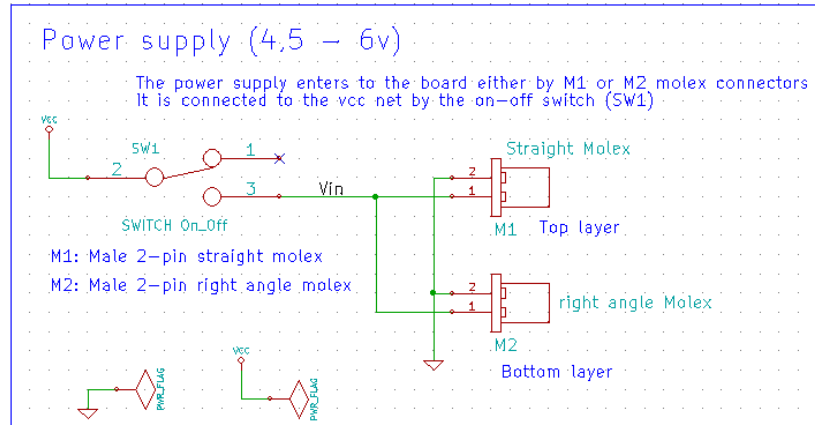


Imagen 15. Detalle esquemático de la alimentación de la placa

LED de encendido

Mediante este LED se comprobará si está encendida o no la placa.

Si el LED está apagado significará que la placa está apagada y si está encendido indicará que la placa está en funcionamiento.

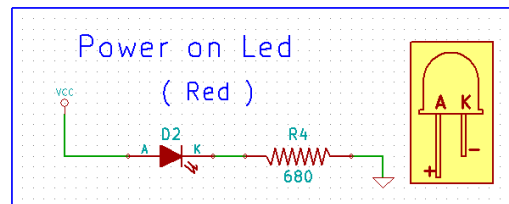


Imagen 16. Detalle esquemático del LED de encendido

Filtro de ruido

Este filtro, como su propio nombre indica, se coloca para reducir el ruido que introduce la fuente de alimentación. El filtrado se encargará de eliminar las componentes de baja y alta frecuencia, evitando posibles perturbaciones en las señales.

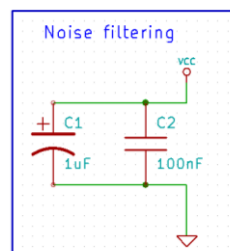


Imagen 17. Detalle esquemático del filtro de ruido

ICSP

El ICSP, que se utiliza para cargar el bootloader al microprocesador consta de 6 pines. El pin 1 va conectado al pin MISO del microprocesador, el pin 2 al PB5 y el pin 5 al MOSI. El pin 6 es la alimentación, el pin 4 se conecta a tierra y el pin 3 se utiliza para resetear.

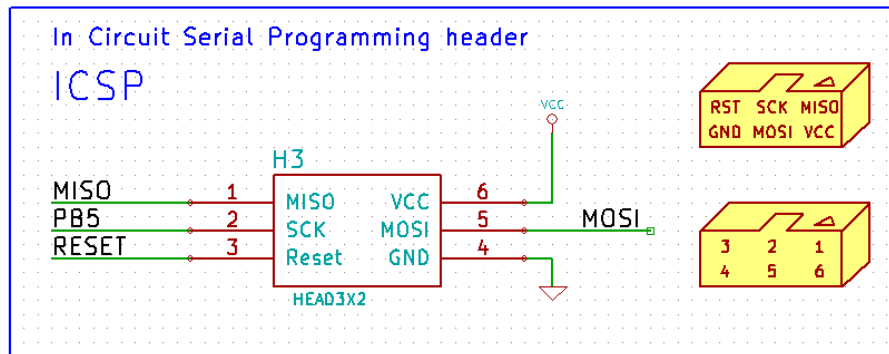


Imagen 18. Detalle esquemático del puerto ICSP

Puerto serie

El puerto serie, como interfaz de comunicación de datos digitales, se usará para establecer comunicación entre la placa y cualquier otro periférico que acepte dicha comunicación.

El pin 1 se conecta a GND, el pin 2 al pin TX del microprocesador, el pin 3 al RX del microprocesador y el pin 4 se utiliza para resetear, según se muestra en la siguiente imagen.

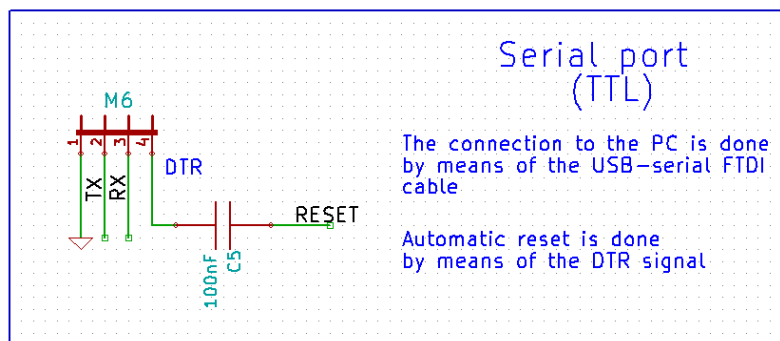


Imagen 19. Detalle esquemático del puerto serie

Conectores de los servos

Se han colocado ocho conectores para los servos, cuatro de ellos se colocarán en la cara superior y cuatro en la cara inferior. Estos conectores tienen tres pines cada uno. El primer pin se conectará a GND, el siguiente a alimentación-VCC- y el último a un puerto digital del microprocesador. En este caso se han utilizado el puerto C (PC0, PC1, PC2 y PC3) y el puerto B (PB0, PB1, PB2, PB3), como se muestra en la siguiente imagen.

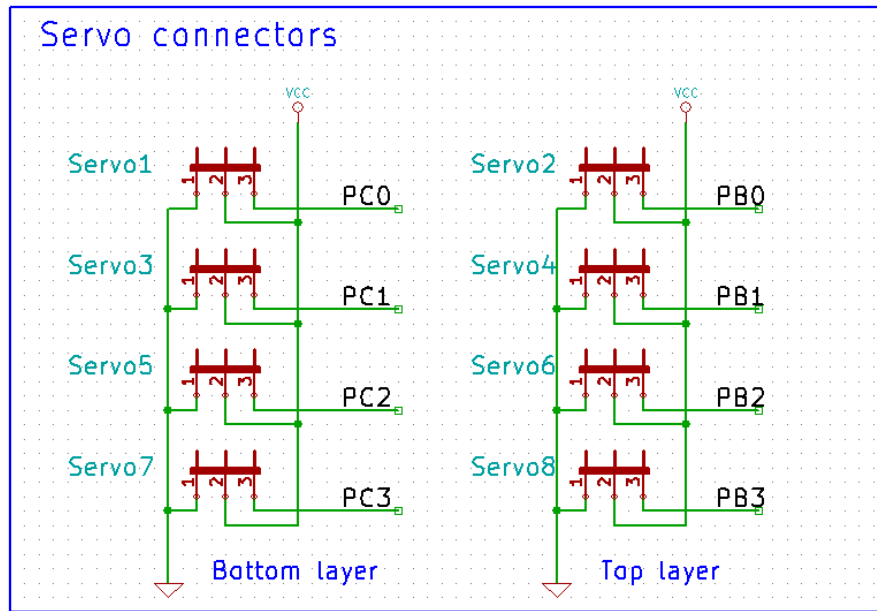


Imagen 20. Detalle esquemático de los conectores de los servos

El uso que se les da a estos conectores, aunque hayan sido diseñados especialmente para la conexión de servos, dependerá de la necesidad del usuario, no queda reducido su uso a este fin.

Puerto de expansión

El puerto de expansión se utiliza para conectar otras placas a la placa Skymega1.0. Consta de diez pines, ocho de los cuales van conectados a puertos del microprocesador, y los dos restantes a alimentación (pin 5) y GND(pin 6).

Los puertos del microprocesador utilizados serán el puerto C y el puerto D (PC0, PC1, PD2, PD3, PD4, PD5, PD6, PD7) según se muestra en la siguiente imagen.

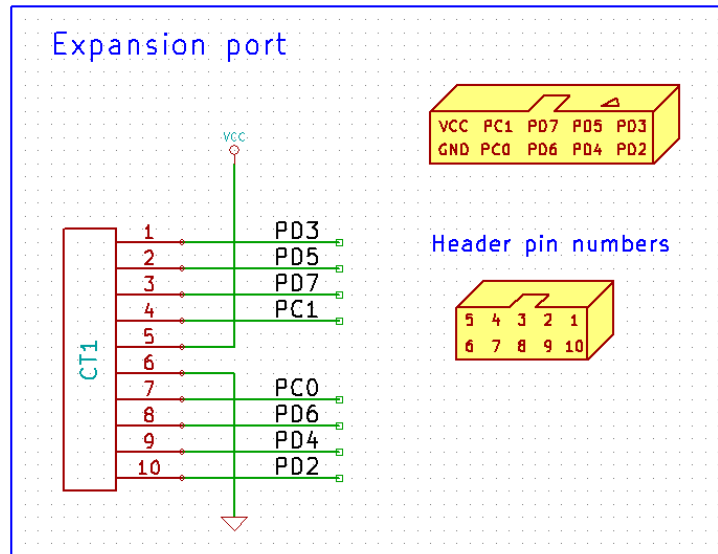


Imagen 21. Detalle esquemático del puerto de expansión

LED de prueba

El LED de prueba se utiliza para probar el software a nivel de salida, facilitando la depuración del mismo. Este LED se conecta al puerto PB5 del microprocesador como se muestra en la siguiente figura.

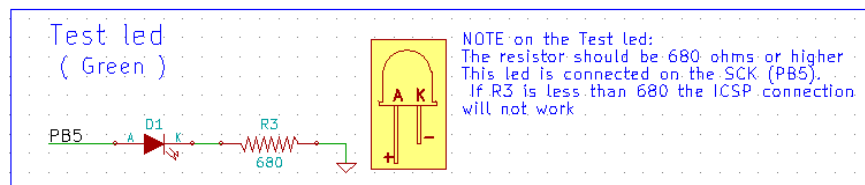


Imagen 22. Detalle esquemático del LED de prueba

Pulsador de prueba

El pulsador de prueba se utiliza para probar el software a nivel de entrada (se usa como entrada digital). Este pulsador se conecta al puerto PB4 del microprocesador como se muestra en la siguiente figura.

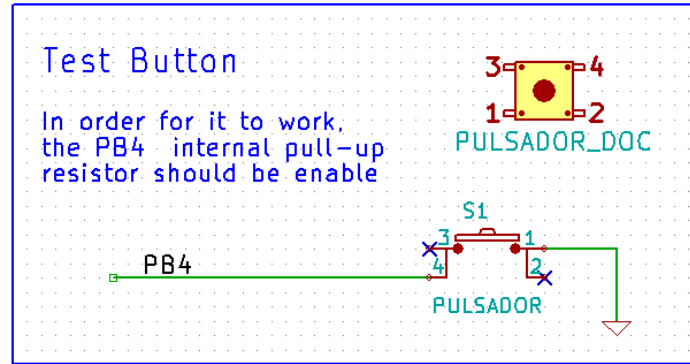


Imagen 23. Detalle esquemático del pulsador de prueba

Bus I2C

El bus I2C es un bus de comunicaciones serie. Por el pin 2, SDA, se transmitirán datos y por el pin 3 SCL, la señal de reloj. Estos dos pines van conectados respectivamente a los pines 27 y 28 del microprocesador. El pin 1 es la alimentación y el pin 4 se conecta a GND. Se han colocado dos Bus I2C para poder conectarlo por un lado o por otro de la placa.

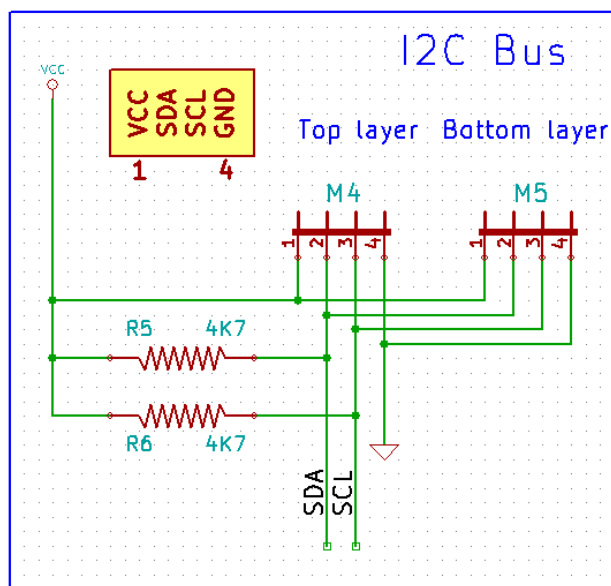


Imagen 24. Detalle esquemático del bus I2C

3.1.4.3. Rutado

Una vez terminado el diseño, se ha realizado el rutado de la placa. Más adelante se explicará cómo se ha llevado a cabo [Manual KiCad]. En la imagen siguiente se muestra el resultado final.

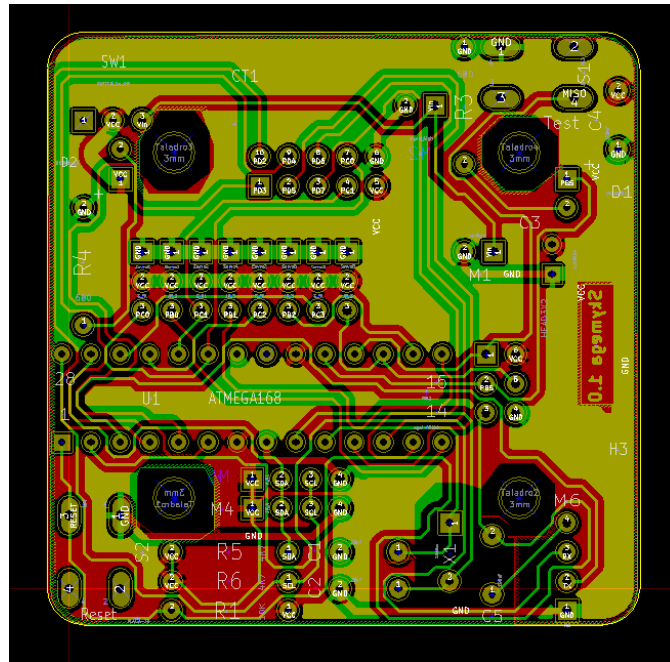


Imagen 25. Skymega1.0 rutada

Archivos de fabricación

Con el rutado de la placa terminado, se han impreso los archivos de fabricación. En la imagen mostrada a continuación se muestran las dos caras de la placa.

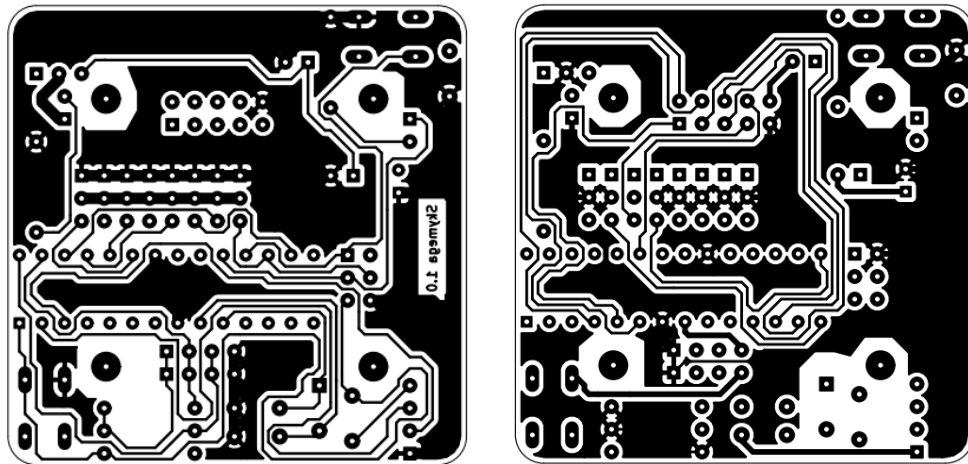


Imagen 26. Archivos fabricación Skymega1.0

3.1.5. Alimentación de la Skymega1.0

Como ya se ha indicado anteriormente, la alimentación de la placa deberá encontrarse en un rango de entre 4,5V y 6V. Se va a alimentar a través de un cable con un conector Molex hembra de dos vías. Este cable deberá fabricarse quedando como se muestra en la siguiente imagen.

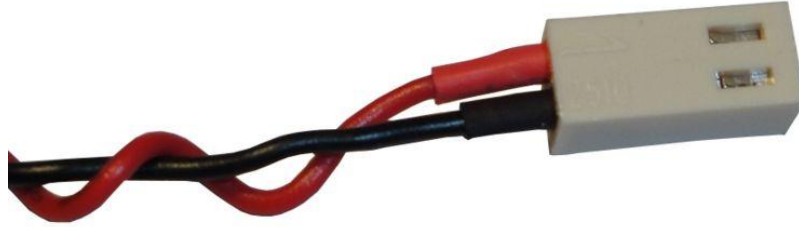


Imagen 27. Cable de alimentación Molex

Con este cable, se alimentará la placa, pudiéndose hacer de varias formas. Una de ellas será mediante un portapilas. La alimentación podrá realizarse de otro modo, siempre que se mantenga el mismo esquema de conexión y los niveles de tensión indicados.



Imagen 28. Portapilas con el conector Molex

El cable de alimentación Molex se conectará a la placa Skymega1.0 como se ve en la siguiente figura.

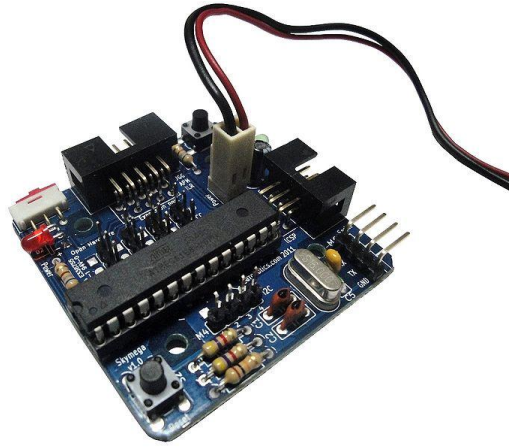


Imagen 29. Conexión del Cable Molex en la Skymega1.0

Un ejemplo, es el mostrado en la siguiente imagen, en el que la alimentación de la placa proviene de un portapilas impreso mediante la impresora 3D.

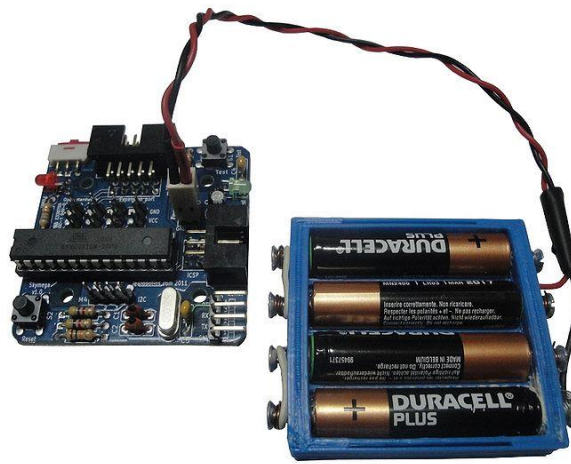


Imagen 30. Alimentación Skymega1.0 con portapilas imprimible

Otro ejemplo de alimentación, es mediante un cable USB-Molex, o por una fuente de alimentación con un cable terminado en conector Molex.

Se muestra a continuación el cable USB-Molex que se debe fabricar.



Imagen 31. Cable USB- Molex

En la siguiente imagen se muestra la alimentación de la placa Skymega1.0 mediante una fuente de alimentación y un conector Molex.

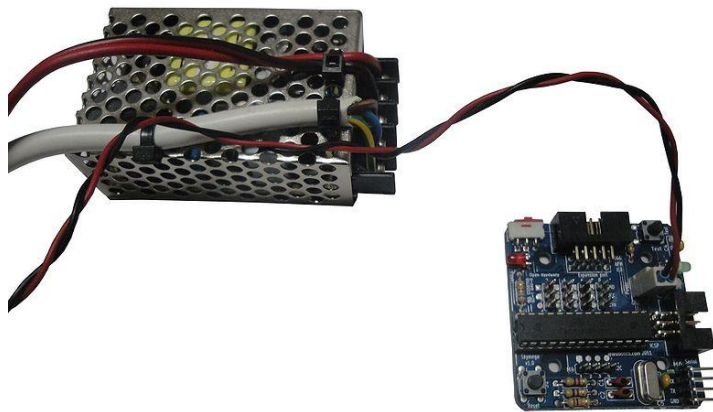


Imagen 32. Alimentación de la placa Skymega1.0 con fuente de alimentación

3.1.6. Carga del bootloader

Cuando el microcontrolador es virgen y no se ha utilizado previamente se deberá cargar en él el bootloader, como se ha indicado anteriormente.

Esto se hará mediante el cargador AVR USBtinyISP2.0.

En la siguiente imagen se puede observar cómo se realiza la conexión de la Skymega1.0 con el programador AVR y el ordenador. La conexión entre el programador y la placa se realizará mediante el puerto ICSP de la placa, del que ya se ha hablado, con un bus de datos. La conexión entre el programador y el ordenador se hará mediante un cable USB.

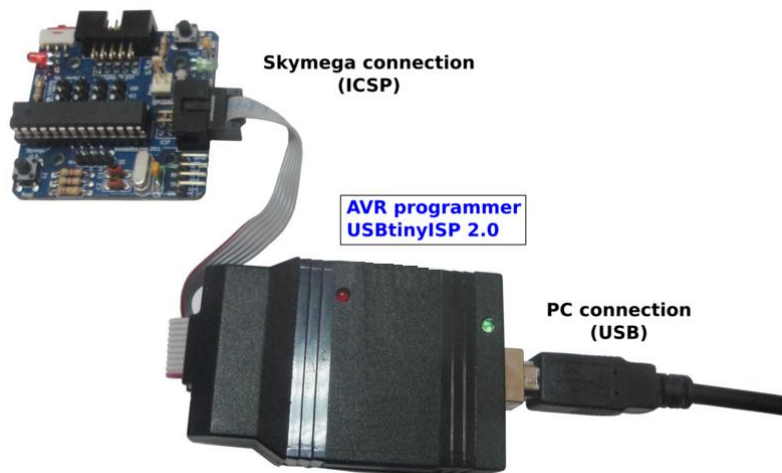


Imagen 33. Conexión de la placa Skymega1.0 con el programador AVR.

A continuación se muestra un detalle de la conexión de la placa con el programador mediante el puerto ICSP de la placa.

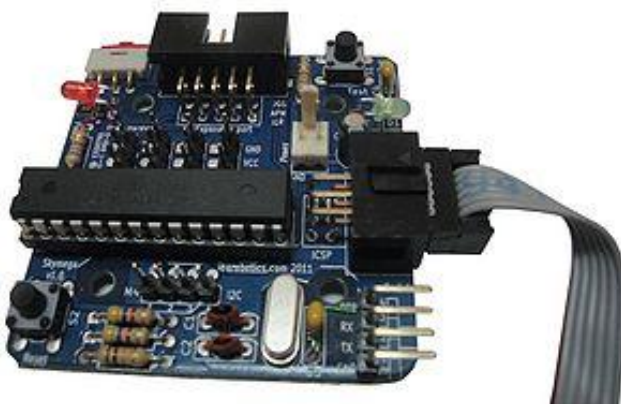


Imagen 34. Detalle de la conexión de la placa Skymega1.0 conector ICSP

3.1.7. Conexión con el PC

La Skymega1.0 se conecta al PC a través de un cable USB-serie de FTDI (modelo TTL-232R-5V). Este cable tiene un conector de seis pines. Para usarlo con la Skymega1.0 es necesario modificar este conector y sustituirlo por uno de cuatro pines como se muestra en las fotos. En el caso que no se disponga del conector de cuatro pines, se puede reutilizar el de seis, sin conectar los dos hilos sobrantes.

Este cable sirve para descargar firmware en la Skymega1.0, así como comunicarse vía puerto serie con ella mediante el ordenador o cualquier otro dispositivo.

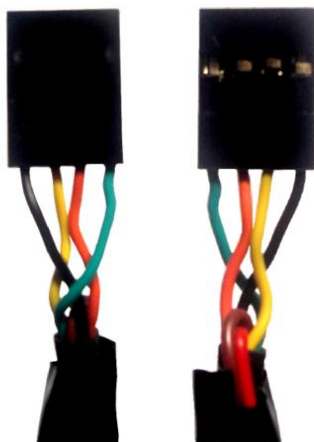


Imagen 35. Detalle Cable USB-serie FTDI

En la siguiente imagen se muestra la conexión de la placa mediante el cable USB-serie de FTDI.

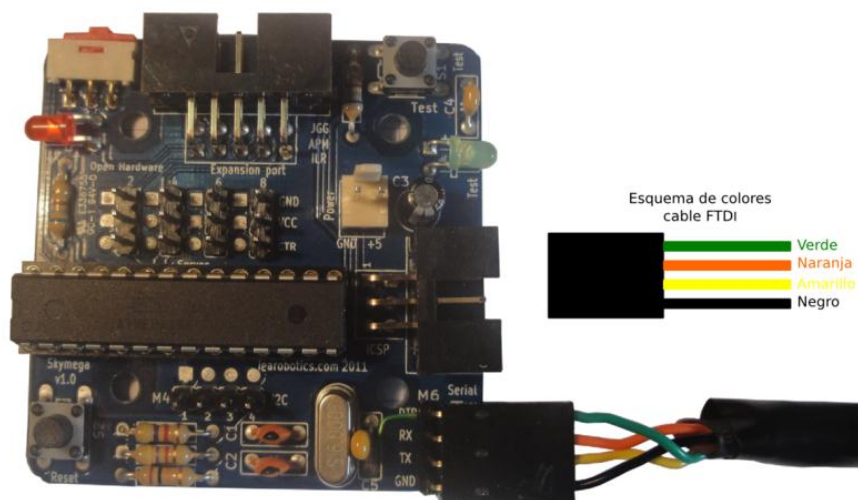


Imagen 36. Conexión Skymega1.0 cable USB- serie FTDI

A continuación, en la siguiente imagen, se muestra el cable USB- serie FTDI utilizado.



Imagen 37. Cable USB-serie FTDI

3.1.8. Conexiones de la placa

3.1.8.1. Conexiones con los servos

En la placa hay ocho conectores diseñados especialmente para servos. La conexión con los servos se hará mediante el siguiente conector, conectándolo directamente al conector de tres pines de la placa. Como se ve en la siguiente figura este conector tiene el pin de GND, el de alimentación-VCC- y un puerto del microprocesador-CTL-.



Imagen 38. Conector del servo Futaba 3003

A continuación se muestra la imagen de un servo Futaba 3003 conectado a la placa Skymega1.0.

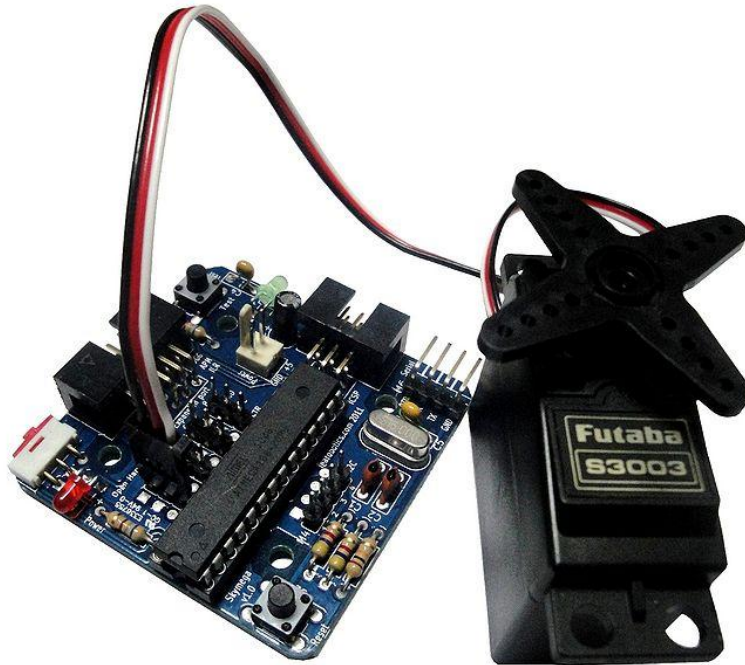


Imagen 39. Servo Futaba 3003 conectado a la placa Skymega1.0

Aunque estos conectores de tres pines han sido diseñados especialmente para conectar servos, pueden ser usados para cualquier otro fin, teniendo en cuenta que el pin que llega desde el microcontrolador a éstos es un pin I/O (pin digital de entrada/salida, o pin analógico de entrada).

3.1.8.2. Conexión del puerto de expansión

Mediante el puerto de expansión se puede conectar la placa Skymega1.0 con otras placas.

Unos ejemplos pueden ser la conexión de la Skymega1.0 con la tarjeta FreeLEDs⁵ o con la tarjeta Skypads⁶, tarjetas realizadas por Juan González Gómez.

La tarjeta FreeLEDs es una placa con ocho LEDs, utilizada para hacer pruebas.

En la siguiente imagen se puede observar la placa Skymega1.0 conectada mediante el puerto de expansión a la placa FreeLEDs.

⁵ La tarjeta FreeLEDs es una placa que consiste en ocho LEDs para hacer pruebas. Ha sido realizada por Juan González Gómez.

⁶ La tarjeta Skypads es una placa que consiste en ocho pulsadores para hacer pruebas. Ha sido realizada por Juan González Gómez.



Imagen 40. Skymega1.0 y FreeLEDs conectadas mediante el puerto de expansión

La tarjeta Skypads es una placa con ocho pulsadores simulando un "gamepad". En la siguiente imagen se puede observar la placa Skymega1.0 conectada mediante el puerto de expansión a la placa Skypads.



Imagen 41. Skymega1.0 y Skypads conectadas mediante el puerto de expansión

3.1.9. Pruebas

3.1.9.1. Pruebas hardware

Los pasos que se han realizado hasta llegar al resultado final han sido los siguientes:

En primer lugar, una vez diseñado el circuito, se ha realizado un prototipo inicial en el que se han soldado los componentes en una placa de puntos, quedando de la siguiente forma.



Imagen 42. Primer prototipo Skymega1.0

Una vez diseñado el circuito en KiCad, se han soldado todos los componentes en una placa de puntos más pequeña que la inicial. Se ha realizado con las mismas dimensiones que la placa Skypic⁷ -realizada por - para que se pueda colocar sobre los robots imprimibles como por ejemplo sobre el Miniskybot, Robot realizado por Juan González Gómez.

En la siguiente imagen se muestra el segundo prototipo de la placa Skymega1.0.



Imagen 43. Segundo prototipo de la Skymega1.0.

⁷ La placa Skypic es una placa realizada por Juan González Gómez y Andrés Prieto-Moreno. La placa Skymega1.0 se ha basado en ella.

Posteriormente, una vez comprobado que la placa funcionaba correctamente se realizó el rutado en KiCad y se mandó fabricar la placa, esta vez, obteniendo una placa de dimensiones mucho menores.

A continuación, en la siguiente imagen, se muestra la cara superior de la placa.

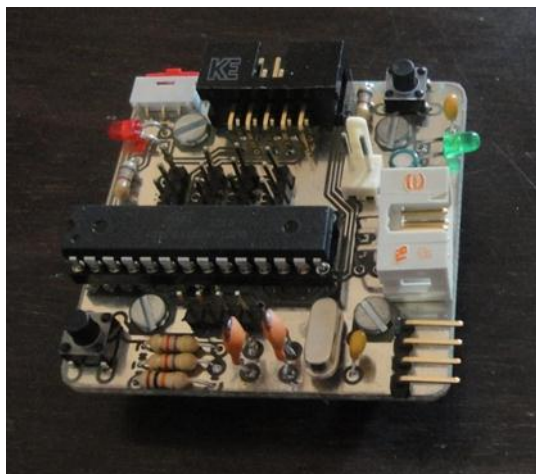


Imagen 44. Tercer prototipo Skymega1.0

3.1.9.2. Pruebas software

Se han realizado una serie de pruebas software para comprobar el correcto funcionamiento de la placa.

Estas pruebas han consistido en cargarle un código al microprocesador y comprobar que la placa funciona correctamente.

Esta placa ha sido posteriormente usada en robots con diferentes funcionalidades.

3.1.10. Resultado final

Después de realizar todas las pruebas hardware y software se mandó fabricar la placa de forma industrial. Se imprimieron cien unidades para poder utilizarlas en los robots Printbots -robots imprimibles-.

El resultado final de la placa es el siguiente:

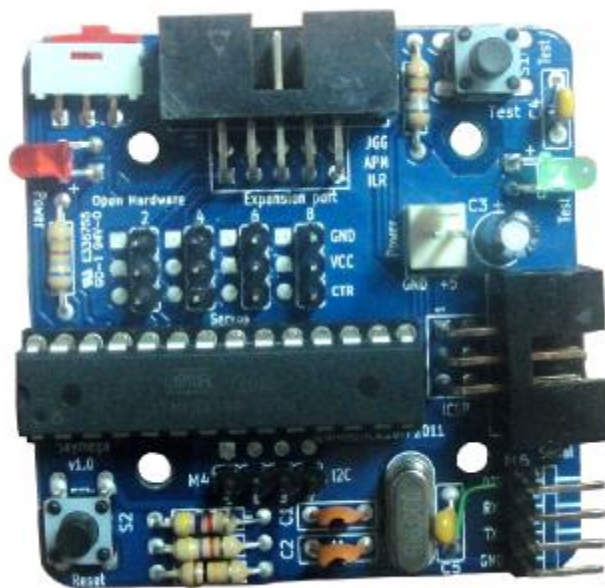


Imagen 45. Placa Skymega 1.0

3.1.11. Mejoras

Atendiendo a las necesidades de los compañeros del grupo de robótica se ha observado que se podría hacer una placa con una mayor funcionalidad que la anteriormente descrita. Se pensó entonces en poner motores de continua con sus respectivos puentes en H para su control –giro y velocidad-, conectores para sensores ultrasonidos e infrarrojos y para un módulo Bluetooth.

En la siguiente tabla se muestran los puntos flojos de la Skymega1.0.

Función	Cumplimiento
Etapas de potencia integrada	NO
Conexión de sensores IR	SI
Conexión de sensores ultrasonido	SI
Control de motores de continua	NO
Conexión de servos	SI
Conexión para bluetooth	NO
Comunicación I2C	SI
Detección de baja tensión para LiPo	NO
Amplio rango tensión entrada	NO

Tabla 2. Puntos débiles Skymega1.0

Debido a las carencias mostradas en la tabla anterior, se diseñó la Skymega2.0 que se explicará a continuación.

3.2 Skymega2.0

3.2.1. Introducción

Para la realización de esta placa se partió de la Skymega1.0. Tras hacer muchas pruebas y conocer las necesidades de los robots móviles de bajo coste, ha sido mejorada creándose así la Skymega2.0.

3.2.2. Uso

Al igual que la Skymega1.0, el uso principal de la placa será el control de robots móviles de bajo coste. Son compatibles además con otras placas y con Arduino.

Aunque ha sido diseñada especialmente para este fin, puede ser usado para cualquier otro, dentro de sus posibilidades y características.

3.2.3. Descripción y Componentes

3.2.3.1. Componentes de la placa

En la siguiente tabla se muestran los componentes para la fabricación de la placa Skymega2.0.

Componente	Valor	Cantidad	Referencia
Condensador cerámico/multicapa	22pF	2	C1, C2
Condensador cerámico/multicapa	100nF	4	C4, C5, C6, C7
Condensador electrolítico	1uF/16v	1	C3
Condensador cerámico/multicapa	100UF	1	C8
Conector acodado 2x5 pines (IDC)	-	1	CT1
LED de 3mm	Verde	1	D1
LED de 3mm	Rojo	2	D2, D3
Conector Molex macho recto de 2 pines	2,54mm	2	CP1, CP2
Conector acodado 2x3 pines (IDC)	-	1	H3
Integrado L293 (puentes en H)	-	1	IC1

Conector Jack de alimentación	-	1	J21
Tira de 3 pines macho rectos	-	7	Servo1, servo2, servo3, servo4, infrared1, infrared2, infrared3.
Tira de 4 pines macho rectos (1x4)	-	4	Ultrasound1, ultrasound2, Bluetooth, M6.
Tira de 4 pines hembra rectos (1x4)	-	1	M4
Resistencia de 1/4W	10K	3	R1,R7,R2
Resistencia de 1/4W	680	1	R3
Resistencia de 1/4W	4K7	2	R6, R5
Resistencia de 1/4W	348	2	R9, R10
Resistencia de 1/4W	1k	1	R8
Resistencia de 1/4W	15K	1	R4
Diodo de polarización inversa	200mA	1	InvPolDiode1
Componente TC54	-	1	U2
Integrado PTH08000WAZT	-	1	U3
Pulsador para circuito impreso	-	2	S1, S2
Interruptor de 2 posiciones acodado para CI	-	3	SW1, SW2, SW3
Cristal de cuarzo	16MHz	1	X1
Zócalo de 2x14 pines (300 mils)	-	1	U1
Microcontrolador Atmega 328P de ATMEL	-	1	U1

Tabla 3. Componentes de la placa Skymega2.0

3.2.3.2. Conexiones con el microprocesador.

Los pines del microprocesador que están disponibles son los mismos que para la Skymega1.0. Esto es debido a que, aunque se utilicen dos microprocesadores diferentes, ambos tienen los mismos puertos disponibles. La única diferencia entre ellos será la memoria disponible para código.

A estos puertos irá conectado lo siguiente:

- Pulsador de prueba
- LED de prueba
- Bus I2C
- Puerto de expansión
- Cuatro servos
- Circuito USB.
- Dos switch para elegir si se quiere utilizar el LED y el pulsador de prueba o el bus I2C
- Dos sensores ultrasonidos (a los puertos PB4, PB5, PD5, PD6)
- Tres sensores infrarrojos (a los puertos (PB2, PB0, PB1))
- Puentes en H
- ICSP
- Conector puerto serie

El *pulsador* y el *LED de prueba* se usarán, como su propio nombre indica, para realizar pruebas a la hora de programar. Se realizarán programas sencillos para comprobar el correcto funcionamiento de la placa.

El *Bus I2C* es un bus de comunicaciones serie. Por el pin 2, SDA, se transmitirán datos y por el pin 3 SCL, la señal de reloj.

El *Puerto de Expansión* se utiliza para conectar otras placas a la placa Skymega2.0.

Esta placa tiene conectores para cuatro servos, tres infrarrojos, y dos ultrasonidos.

Los cuatro servos van conectados a GND, alimentación y a un puerto del microcontrolador (PC0, PC1, PC2, PC3).

Los tres sensores *infrarrojos* van conectados también a GND, alimentación y a un puerto del microcontrolador. En este caso se utilizarán los puertos PB0, PB1 y PB2.

Los dos sensores *ultrasonidos* que se han propuesto, van conectados a GND, alimentación, y a dos puertos del microcontrolador cada uno. Los puertos utilizados han sido PB4, PB5, PD5 y PD6.

Los dos *switch*, *Switch PC4* y *Switch PC5*, se utilizarán para elegir si se quiere utilizar el LED y el pulsador de prueba o el bus I2C. Se ha realizado de este modo el

diseño de la placa porque se ha estimado que el LED y el pulsador de prueba no se utilizarán más que en determinadas ocasiones. De este modo, se ha conseguido tener libres dos puertos más del microprocesador, aumentando la capacidad de la placa en cuanto a sensores y funcionalidad.

Los *puentes en H* se utilizan para controlar los motores de continua. Mediante el integrado L293 (dos puentes en H en el mismo integrado) se puede controlar cuatro motores que giren en un sentido, o bien dos motores que giren en los dos sentidos, que es en este caso lo que se busca. Para ello se conectará el integrado a cuatro puertos del microcontrolador (dos PWM y dos puertos digitales). Mediante el pin digital se controla el sentido de giro y con el PWM la velocidad de giro.

El *puerto ICSP* sirve para cargar el bootloader al microprocesador. Esto se realizará una única vez, cuando el microcontrolador es virgen. Si ya se ha cargado previamente no hay que volver a cargarlo.

A través del *Puerto Serie* se le cargarán al microprocesador los programas.

El *Bluetooth* irá conectado a los puertos TX y RX del microcontrolador. Hay que tener en cuenta que el puerto serie y el Bluetooth no podrán conectarse al mismo tiempo. Esto no será un problema porque normalmente cuando se carga el código al robot, éste no utiliza el Bluetooth.

3.2.4. Diseño de la placa

El diseño de la placa se ha realizado mediante el programa libre KiCad.

Se ha realizado un esquemático y posteriormente se ha rutado. A continuación se explicarán los pasos para la realización del esquemático y del rutado de la placa.

3.2.4.1. Esquemático

El esquemático de la placa Skymega2.0 es el siguiente:

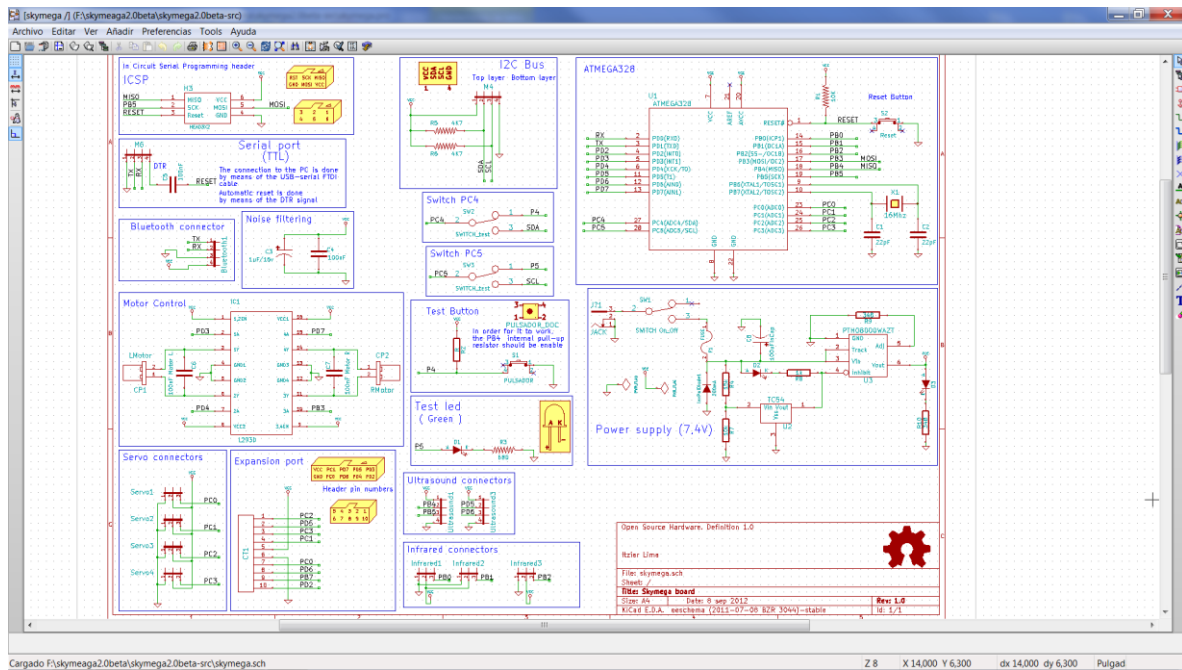


Imagen 46. Esquemático Skymega2.0

3.2.4.2. Criterios de diseño

A continuación se explicará el esquemático detalladamente.

Microprocesador

El microprocesador utilizado será el Atmega328P. En la imagen siguiente se pueden observar las conexiones de éste.

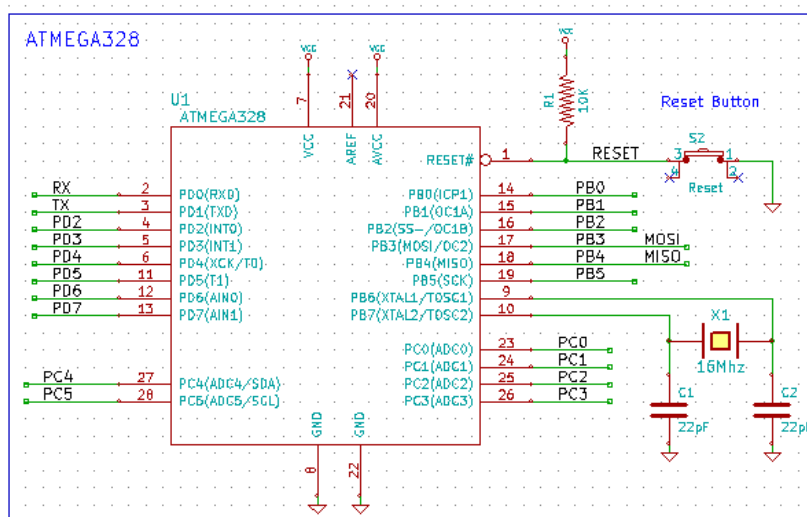


Imagen 47. Detalle esquemático del Atmega328

El cristal de 16MHz se conectará a los pines 9 y 10.

El pin 1 es el pin de Reset, al que se le conectará un pulsador mediante el cual se podrá resetear el microprocesador.

El puerto B (PB7... 0) es un puerto I/O de 8 bits bidireccional.

El puerto C (PC5... 0) es un puerto I/O de 6 bits bidireccional.

El puerto D (PD7...0) es un puerto I/O de 8 bits bidireccional.

Más adelante se describirán las conexiones con estos puertos.

Alimentación, circuito de low voltage, LED de encendido y regulador de tensión

- Alimentación

La alimentación, que se realizará a través de un conector jack, deberá estar comprendida entre 7,2V y 14V, un rango muy superior al aceptado por la Skymega1.0.

La alimentación en esta placa es algo crítico e importante y se hablará más adelante del porqué.

Se ha colocado un interruptor para que de este modo se pueda encender y apagar la placa.

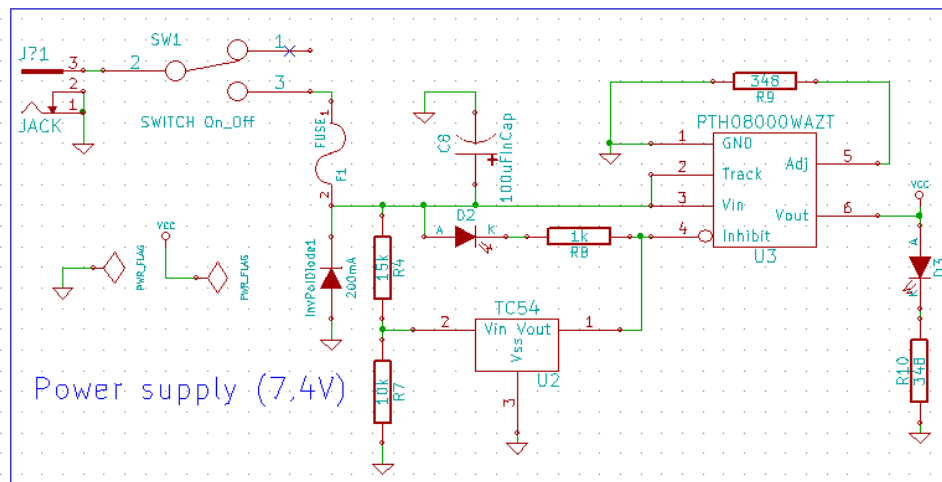


Imagen 48. Detalle esquemático de la alimentación de la placa

- Circuito de Low Voltage

El integrado TC54 tiene como función evitar que la tensión baje por debajo de 7,2V.

El uso de este componente es debido a que la placa ha sido diseñada especialmente para ser alimentada con una batería LiPo, debido a las grandes

ventajas que aportan: grandes corrientes, posibilidad de dar picos de corrientes durante cortos periodos de tiempo –del orden de ms-, facilidad y rapidez de carga, duración, tamaño, etc.

A pesar de las grandes ventajas que éstas presentan, hay que tener cuidado a la hora de usarlas, ya que si se les somete a descargas muy profundas, la vida útil de estas baterías se reduce considerablemente. Es por esto por lo que se ha colocado un integrado que se encargará de monitorizar en tiempo real el nivel de tensión en la batería, de tal forma que cuando éste se encuentre por debajo de 7,2V la placa se apague, evitando así una descarga profunda.

Junto al detector de “low voltage” existe un LED que nos indicará si la batería está agotada, y por tanto que la placa no está funcionando.

- **Regulador**

El regulador de tensión PTH08000WAZT se encarga de adaptar el nivel de tensión de entrada a un nivel de 5V.

Aunque este regulador acepta un rango de entrada de 5,5V a 14V, debido a lo explicado anteriormente –el uso del TC54- el rango de alimentación de la placa deberá ser de 7,2V a 14V.

- **LED de encendido**

Mediante este LED se comprobará si está encendida o no la placa.

Si el LED está apagado significará que la placa está apagada y si está encendido indicará que la placa está en funcionamiento.

Filtro de ruido

Se coloca un filtrado por medio de un condensador electrolítico y uno cerámico, para evitar ruidos provenientes de la fuente de alimentación, al igual que en la Skymega1.0.

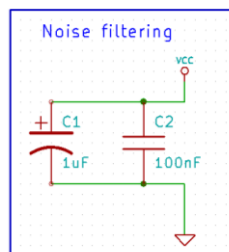


Imagen 49. Detalle esquemático del filtro de ruido

Conectores de los servos

Se han colocado cuatro conectores para los servos. Estos conectores tienen tres pines cada uno. El primer pin va conectado a GND, el siguiente a alimentación y último a un puerto del microprocesador. En este caso se ha utilizado el puerto C (PC0, PC1, PC2 y PC3) como se muestra en la siguiente imagen.

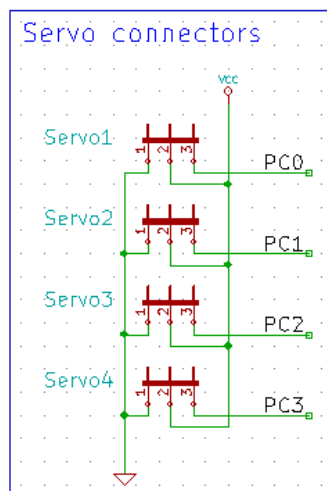


Imagen 50. Detalle esquemático de los conectores de los servos

El uso que se les da a estos conectores, aunque hayan sido diseñados especialmente para la conexión de servos, dependerá de la necesidad del usuario, no queda reducido su uso a este fin.

Puerto de expansión

El puerto de expansión se utiliza para conectar otras placas a la placa Skymega2.0. Consta de diez pines, ocho de los cuales van conectados a puertos del microprocesador, y los dos restantes a alimentación (pin 5) y GND (pin 6).

Los puertos del microprocesador utilizados serán el puerto C, el puerto B y el puerto D (PC0, PC1, PC2, PC3, PB7, PD2, PD5, PD6) según se muestra en la siguiente imagen.

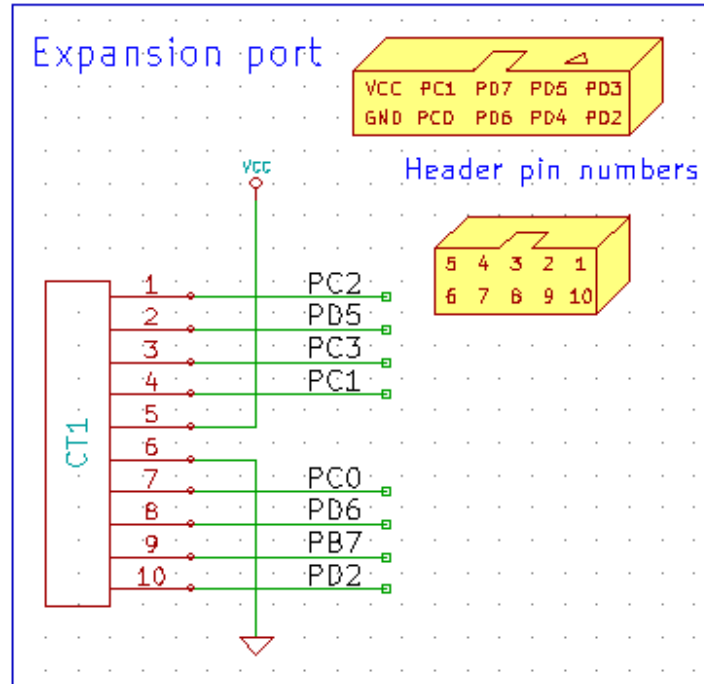


Imagen 51. Detalle esquemático del puerto de expansión

Conectores de los sensores ultrasonido

Se han colocado dos conectores para sensores de ultrasonido. Estos conectores constan de cuatro pines, el primero se conecta a alimentación, los dos siguientes van conectados a puertos del microprocesador, y el último a GND.

Los pines del microprocesador utilizados han sido el puerto B y el puerto D (PB4, PB5, PD5, PD6) como se muestra en la siguiente figura.

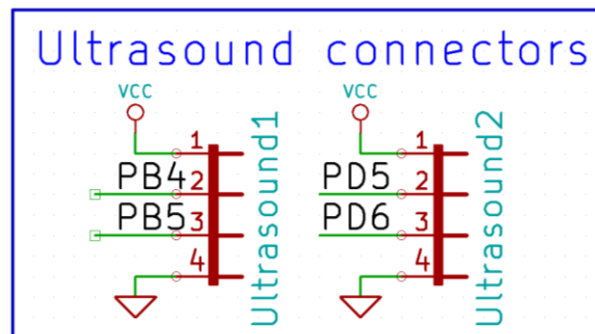


Imagen 52. Detalle esquemático de los conectores de los sensores ultrasonido

El uso que se les da a estos conectores, aunque hayan sido diseñados especialmente para la conexión de sensores ultrasonido, dependerá de la necesidad del usuario, no queda reducido su uso a este fin.

Conectores de los sensores de infrarrojo

Se han colocado tres conectores para sensores de infrarrojo. Estos conectores constan de tres pines. El primero se conecta a GND, el segundo a alimentación y el siguiente a un puerto del microprocesador.

En este caso se ha utilizado el puerto B del microprocesador (PB0, PB1, PB2) según se muestra en la siguiente figura.

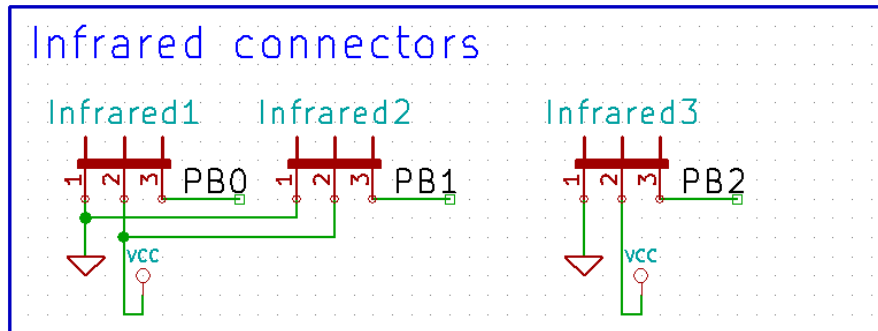


Imagen 53. Detalle esquemático de los conectores de los sensores infrarrojo

El uso que se les da a estos conectores, aunque hayan sido diseñados especialmente para la conexión de sensores infrarrojo, dependerá de la necesidad del usuario, no queda reducido su uso a este fin.

Interruptor Switch PC4

Se han colocado dos interruptores para poder conectar al mismo puerto del microprocesador dos puntos diferentes. En este caso el pin SDA del bus I2C y el pulsador de prueba. Esto se ha hecho porque el Bus I2C y el pulsador de prueba no se utilizan al mismo tiempo pero si comparten un mismo pin del microcontrolador. Este interruptor se ha conectado al puerto PC4 del microprocesador.

Las conexiones se pueden ver en la siguiente imagen.

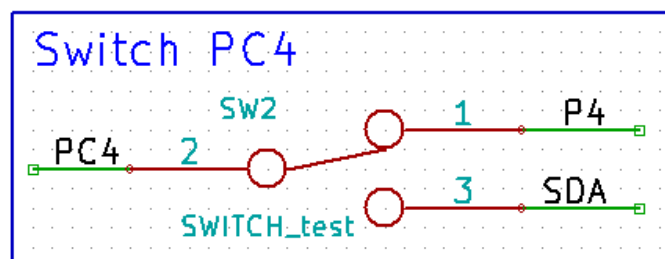


Imagen 54. Detalle esquemático del interruptor Switch PC4

Interruptor Switch PC5

En este caso el pin SCL del bus I2C (SCL) y el LED de prueba (P5). Esto se ha hecho porque el Bus I2C y el LED de prueba no se utilizan al mismo tiempo, pero comparten un mismo pin. Este interruptor se ha conectado al puerto PC5 del microprocesador.

Las conexiones se pueden ver en la siguiente imagen.

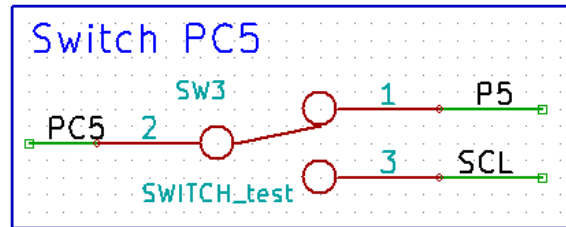


Imagen 55. Detalle esquemático del interruptor Switch PC5

LED de prueba

El LED de prueba se utiliza para hacer las pruebas software. Este LED se conecta al interruptor SWPC2 como se muestra en la siguiente figura.

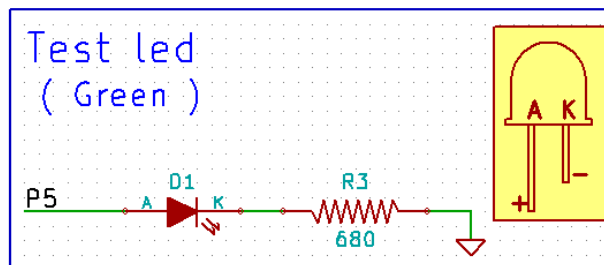


Imagen 56. Detalle esquemático del LED de prueba

Pulsador de prueba

El pulsador de prueba se utiliza para hacer las pruebas software. Este pulsador se conecta al interruptor SWPC1 como se muestra en la siguiente figura.

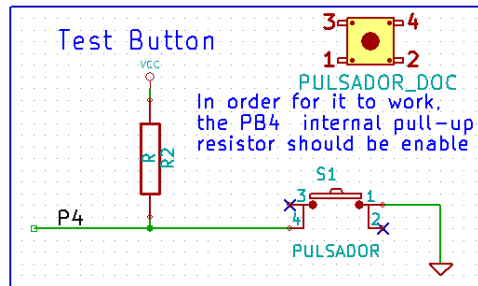


Imagen 57. Detalle esquemático del pulsador de prueba

ICSP

El ICSP, que se utiliza para cargar el bootloader al microprocesador consta de 6 pines. El pin 1 va conectado al pin MISO del microprocesador, el pin 2 al PB5 y el pin 5 al MOSI. El pin 6 es la alimentación -VCC-, el pin 4 se conecta a GND y el pin 3 se utiliza para resetear -RESET-, como se muestra en la siguiente imagen.

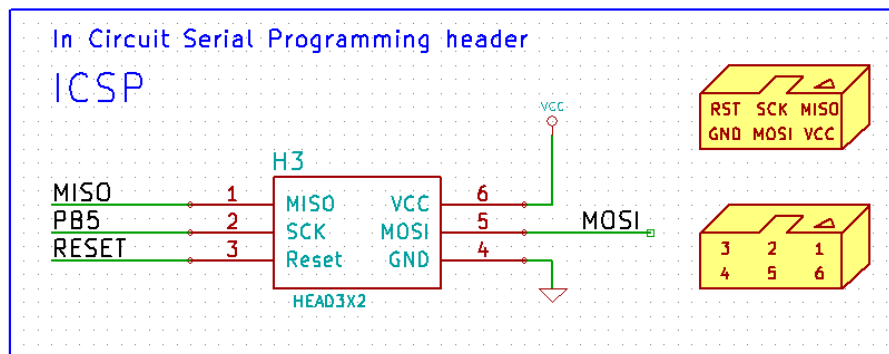


Imagen 58. Detalle esquemático del puerto ICSP

Puerto serie

Al igual que en la Skymega1.0, el puerto serie, como interfaz de comunicación de datos digitales, se usará para establecer comunicación entre la placa y cualquier otro periférico que acepte dicha comunicación.

El pin 1 se conecta a GND, el pin 2 al pin TX del microprocesador, el pin 3 al RX del microprocesador y el pin 4 se utiliza para resetear -RESET-, según se muestra en la siguiente figura.

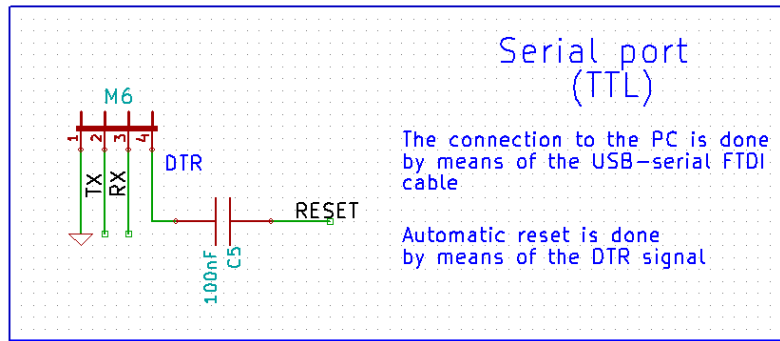


Imagen 59. Detalle esquemático del puerto serie

Bus I2C

El bus I2C es un bus de comunicaciones serie. Por el pin 2, SDA, se transmitirán datos y por el pin3 SCL, la señal de reloj. Estos dos pines van conectados respectivamente al interruptor SWPC1 y al SWPC2. El pin 1 es la alimentación – VCC- y el pin 4 se conecta a GND.

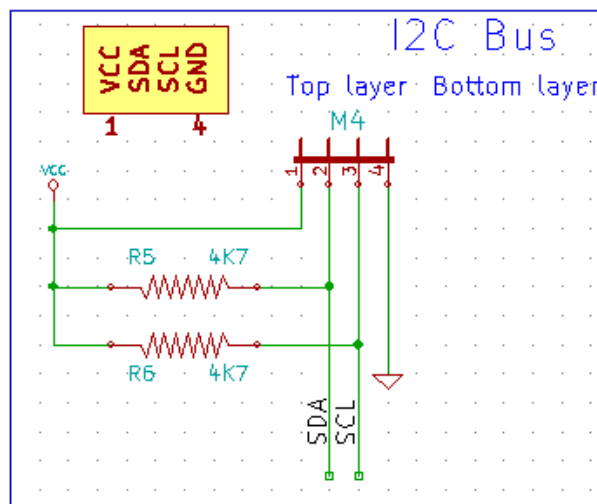


Imagen 60. Detalle esquemático del bus I2C

Conector Bluetooth

El conector Bluetooth irá conectado al microcontrolador a través de los puertos TX y RX. El pin TX del Bluetooth deberá ir conectado al puerto RX del microprocesador y el pin RX del Bluetooth deberá conectarse al puerto TX del microprocesador.

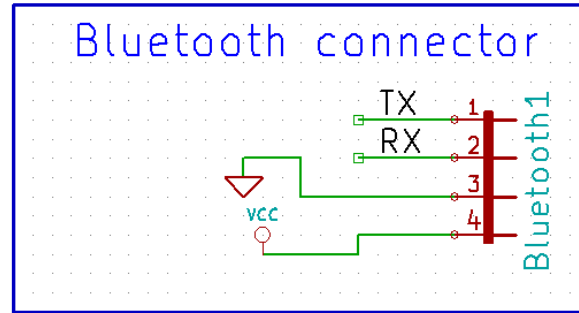


Imagen 61. Detalle esquemático del conector Bluetooth

Control del Motor

El control de los motores de continua se realiza mediante los puentes en H.

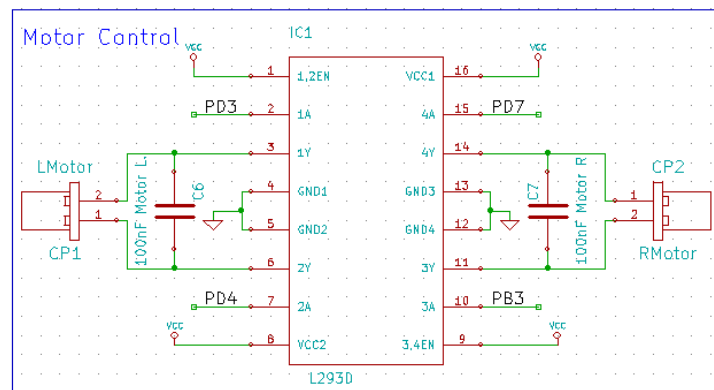


Imagen 62. Detalle esquemático control del motor

- Puentes en H

Mediante el integrado L293 se pueden controlar cuatro motores que giren en un sentido, o bien dos motores que giren en los dos sentidos, que es en este caso lo que se quiere. Para ello se conectará el integrado a cuatro pines del microcontrolador (dos PWM y dos pines digitales). Mediante el pin PWM se modificará la velocidad y mediante el digital se variará el sentido de giro.

- Conectores de los motores

Los conectores de los motores son dos conectores Molex macho. Cada uno de estos conectores corresponde a un motor y va conectado a dos pines del integrado L293.

3.2.4.3. Rutado

Se ha realizado el rutado de la placa como se ha indicado en el manual de KiCad [Capítulo 4].

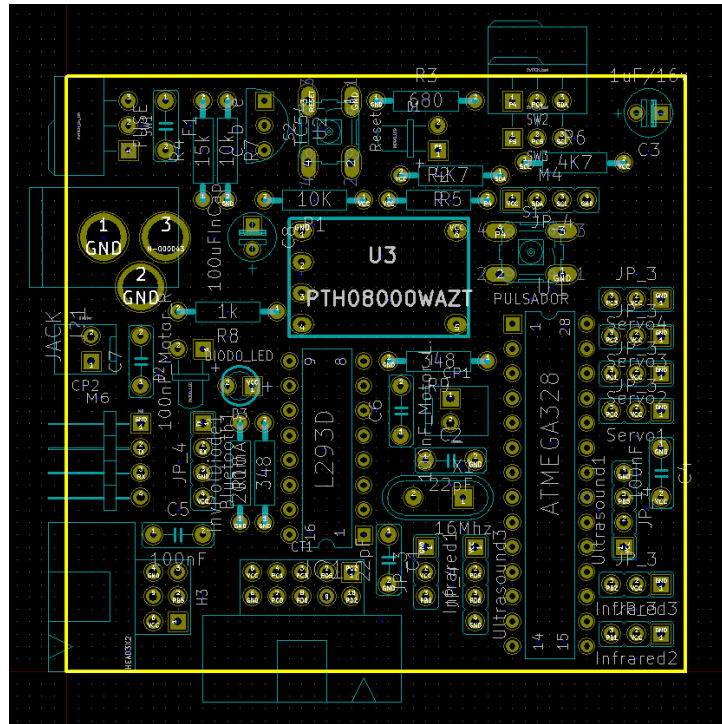


Imagen 63. Colocación componentes en la placa Skymega2.0

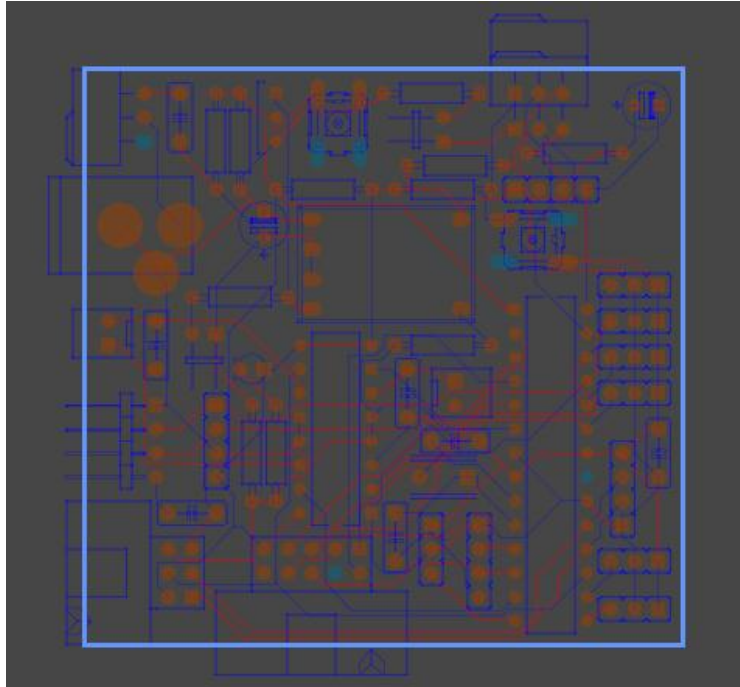


Imagen 64. Placa Skymega2.0 rutado en Specctra

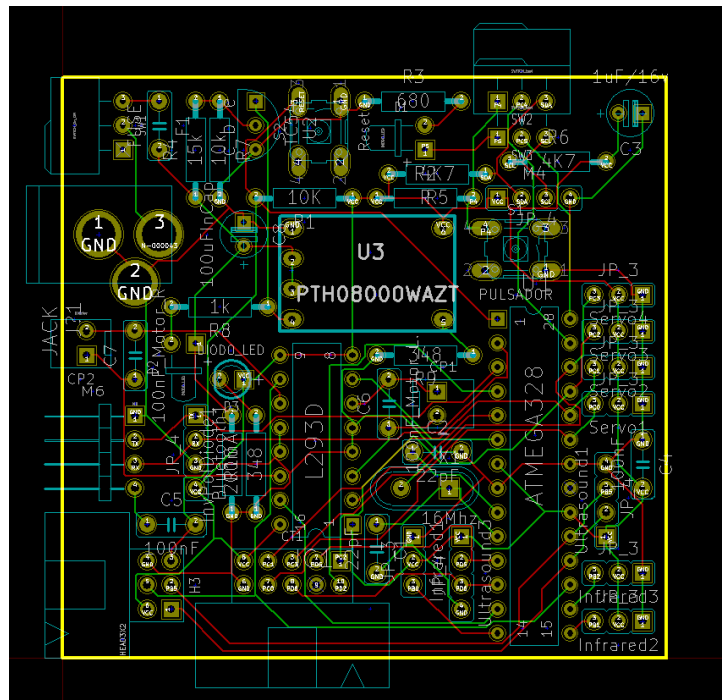


Imagen 65. Placa Skymega2.0 ya rutada e importada a KiCad

Archivos de fabricación

Una vez rutada la placa, se han impreso los archivos de fabricación de ésta. A continuación se muestran las dos caras de la placa.

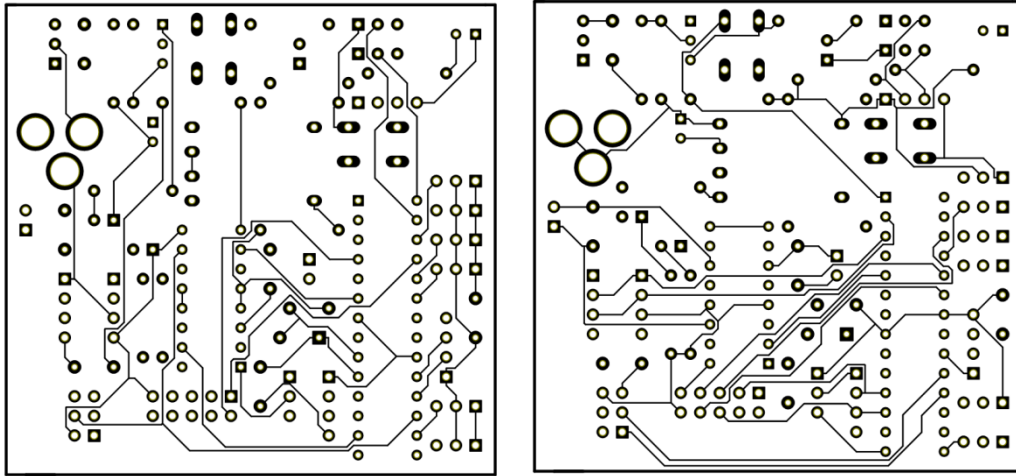


Imagen 66. Archivos de fabricación placa Skymega2.0

3.2.5. Alimentación de la Skymega2.0

La alimentación, de la que ya se ha hablado anteriormente, se realizará a través de un conector Jack como el que se ve en la siguiente imagen.

Seguidamente se coloca un interruptor que permitirá encender y apagar la placa sin necesidad de desconectar la fuente de alimentación.



Imagen 67. Conector Jack

La alimentación en esta placa es algo que adquiere un gran peso. Esto es debido a que el número de componentes que pueden estar funcionando a la vez, tanto de la propia placa como externos a ella, puede ser elevado.

Debido a que existen componentes que tienen un elevado consumo, se ha decidido elegir un convertidor DC-DC conmutado, el PTH08000WAZT del que ya

hemos hablado, que es capaz de dar hasta 2,25A, una corriente que también puede aportar una batería LiPo⁸, de aquí su elección.



Imagen 68. Batería LiPo de 7,4V (dos celdas⁹)

Se ha colocado un interruptor para que de este modo se pueda encender y apagar la placa.

Mediante el LED de encendido se comprobará si está encendida o no la placa.

Si el LED está apagado significará que la placa está apagada y si está encendido indicará que la placa está en funcionamiento.

3.2.6. Carga del bootloader

La carga del bootloader se realiza a través del puerto ICSP de la placa, del mismo modo que en la placa Skymega1.0. Para ver con más detalle la carga del bootloader, se debe acudir al apartado [Carga del bootloader].

3.2.7. Conexión con el PC

La carga del código se realiza a través del Puerto serie de la placa. Se ha introducido una novedad respecto a la Skymega1.0.

⁸ Las baterías **LiPo** (Lithium Polymer) son una línea de evolución de las baterías Li-Ion, en las que se ha sustituido el electrolito líquido orgánico por un compuesto sólido

⁹ La imagen corresponde a una batería *Turnigy 2200 mAh 2S Lipoly Pack*.

En este caso se utilizará un módulo FTDI comprado y un cable USB (de USB a micro USB), en lugar de utilizar un cable con el chip FTDI integrado. Esto se ha decidido así debido a que se ha realizado un estudio de precios, y de esta forma es más económico.

El módulo¹⁰ FTDI propuesto es el siguiente:

Este módulo incluye además el cable USB. Su coste es de 6€ aproximadamente.



Imagen 69. Módulo FTDI

Se pensó también en realizar la placa, pero no salía rentable, debido a que este módulo FTDI (con el cable incluido en el pack) es muy económico.

Este módulo va conectado al puerto serie de la placa mediante un conector de 4 pines. El módulo contiene un conector micro-USB y se conectará con el ordenador mediante un cable USB a micro USB corriente.

Esto hace que descienda el coste respecto a la forma propuesta de conectar la placa Skymega1.0 con el PC.

3.2.8. Conexiones de la placa

3.2.8.1. Conexiones con los servos

La placa Skymega2.0 tiene cuatro conectores para servos. Estos conectores son iguales que los de la placa Skymega1.0 y se conectan de la misma forma. Para mayor información acudir al apartado [Conexiones con los servos] de la placa Skymega1.0.

3.2.8.2. Conexión del Puerto de expansión

La conexión del puerto de expansión se realiza de la misma forma que la Skymega1.0. Para mayor información acudir al apartado [Conexión del puerto de expansión] de la placa Skymega1.0.

¹⁰ El módulo se ha encontrado en la página web DealExtreme: <http://www.dealextreme.com/p/ftdi-basic-breakout-arduino-usb-to-ttl-upload-tool-for-mwc-black-142041?item=1>.

3.2.8.3. Conexión sensores ultrasonido

Los sensores de ultrasonido se conectarán con la placa mediante el conector de cuatro pines. Dos de estos pines irán a un puerto del microcontrolador y los otros son de alimentación y GND. La placa Skymega2.0 consta de dos conectores para sensores de ultrasonido.

3.2.8.4. Conexión sensores infrarrojo

Los sensores infrarrojos se conectarán con la placa mediante el conector de tres pines. Uno de estos pines irá a un puerto del microcontrolador y los otros son de alimentación y GND. La placa Skymega2.0 tiene tres conectores para sensores de infrarrojo.

3.2.8.1. Conexión con los motores de continua

La conexión con los motores de continua se realiza a través de dos conectores Molex macho situados en la placa.

3.2.8.2. Conexión con el módulo Bluetooth

La conexión con el módulo Bluetooth se realiza mediante un conector hembra de cuatro pines que irá situado en la placa.

3.2.9. Pruebas

3.2.9.1. Pruebas hardware

Debido a que parte de la placa es similar a la Skymega1.0, se han probado por separado las partes que difieren de esta para así comprobar que funcionan correctamente.

Posteriormente, una vez diseñado el circuito en KiCad se han soldado todos los componentes en una placa de puntos de dimensiones similares a las de la placa rutada, aunque un poco mayores para mayor comodidad al soldar. Los componentes han sido colocados de igual forma que en el diseño en Kicad.

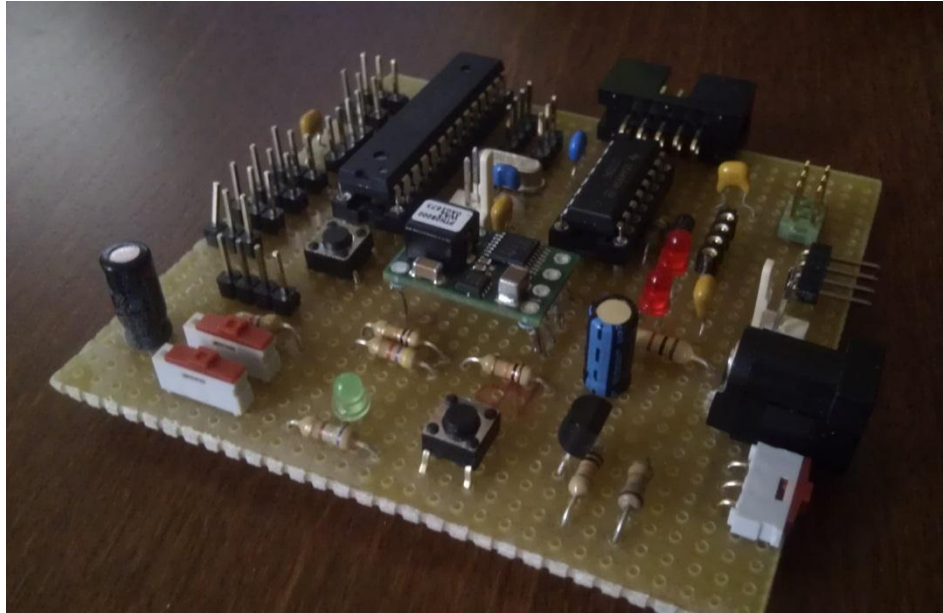


Imagen 70. Placa prototipo Skymega2.0

Se ha intentado realizar un segundo prototipo, fabricando la placa en la Universidad, pero no ha sido posible por falta de tiempo.

Se pidió entonces presupuesto a una empresa externa, pero debido al alto coste de las placas no ha sido posible fabricarla finalmente.

3.2.9.2. Pruebas software

Se han realizado una serie de pruebas software para comprobar el correcto funcionamiento de la placa.

Estas pruebas han consistido en cargarle un código al microprocesador y comprobar que la placa funciona correctamente.

3.2.10. Resultado final

Como resultado final se ha conseguido realizar una placa, con más funcionalidades y mayor capacidad que la Skymega1.0.

A continuación se realizará una comparativa de funcionalidades de las dos placas.

3.3 Comparativa entre las dos placas

En este punto se va a proceder a describir las diferencias entre las dos placas diseñadas, Skymega1.0 y Skymega2.0.

3.3.1. Comparativa de funcionalidades

En la siguiente tabla se muestran las diferentes características que poseen ambas placas. Se ha realizado en modo de tabla para que sea más sencillo comparar las dos placas realizadas.

	Skymega1.0	Skymega2.0
Conectores servos	8	4
Conectores infrarrojo	-	3
Conectores ultrasonido	-	2
Conectores Motores C.	-	2
Puerto de expansión	1	1
LED de prueba	1	1
Pulsado de Prueba	1	1
Bus I2C	1	1
Puerto ICSP	1	1
Puerto Serie	1	1
Conector Bluetooth	-	1
ATMEGA168	1	-
ATMEGA328	-	1

Tabla 4. Comparativa de funcionalidades entre Skymega1.0 y Skymega2.0

Un gran cambio realizado ha consistido en la introducción, en la Skymega2.0, de la electrónica necesaria para poder conectar dos motores de continua. En la placa Skymega1.0, debido a la no existencia de la electrónica para el control de motores de continua, si se requería el uso de estos motores, se sustituirán por servos “trucados”, que consiste en quitar las restricciones mecánicas y electrónicas que posee un servomotor para permitirle girar libremente.

Además de los dos conectores para los motores de continua, se han mantenido cuatro conectores para servos. Estos conectores pueden usarse también para conectar sensores de infrarrojo, en el caso de que hagan falta más sensores de este tipo, ya que se conectan de igual manera (pin de GND, alimentación, puerto del microprocesador).

Se han introducido en la Skymega2.0 dos conectores de ultrasonido que no estaban en la Skymega1.0.

Además otra mejora importante en la Skymega2.0 es el conector Bluetooth, que no poseía la Skymega1.0.

Y por último, otra mejora de la que ya se ha hablado, es la alimentación, pensada para una batería LiPo, junto con toda la electrónica que esto conlleva y la ampliación de rango de tensiones de entrada.

3.4 Trabajos futuros

Se han pensado una serie de mejoras que se podrían realizar en la placa Skymega2.0.

- Mejorar la carga del bootloader y del código en el microprocesador.
- Añadir un conector para la conexión de un módulo XBee.
- Posibilidad de conectar una LiPo de más celdas.

3.4.1. Mejora de la carga del bootloader y del código en el microprocesador.

En las placas Skymega1.0 y en la Skymega2.0 se utiliza el *puerto ICSP* para cargar el bootloader al microprocesador la primera vez que se usa, y posteriormente se deben cargar los códigos al microprocesador a través del puerto serie. Esta mejora de la placa consistiría en que estas dos funciones se realizaran a través del USB¹¹. De esta forma se ahorra espacio y dinero y se simplifica la placa. Ya que en vez de tener dos conectores para hacer estas dos funciones (ICSP para cargar el bootloader y puerto serie para cargar el código), se utiliza únicamente el conector micro-USB.

En la siguiente imagen se muestra el esquemático realizado en KiCad del circuito propuesto.

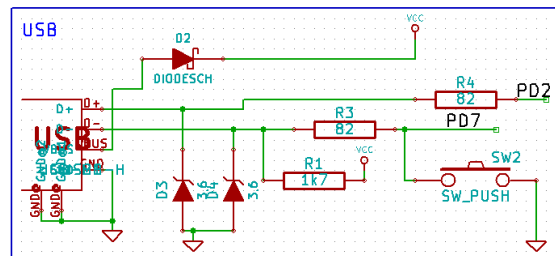


Imagen 71. Detalle esquemático del circuito USB

Además, en lugar de tener que comprar el cable FTDI, que tiene un precio alto, y tener que adaptarlo, se puede usar un cable USB a micro-USB que actualmente es muy común y esto reduciría el precio de la placa.

¹¹ Esta idea se ha extraído de una placa llamada Snoobie.

Hay que tener en cuenta que el cristal necesario debe ser de 12MHz y los condensadores de 27pF. Esto es importante ya que las placas realizadas el cristal que tenía el microprocesador era de 16MHz y los condensadores de 22pF.

Los componentes que se usarían para el circuito del USB serían los siguientes:

Componente	Valor	Cantidad	Referencia
Resistencia de 1/4W	82	2	R4, R3
Resistencia de 1/4W	1k7	1	R1
Diodo Zener 3,6V	-	2	D3, D4
Diodo de protección	-	1	D2
Conector micro USB hembra	-	1	J1
Pulsador para circuito impreso	-	2	SW2, SW3,

Tabla 5. Componentes del circuito USB

Atendiendo al coste de los materiales para la programación actual de las Skymega – 1.0 y 2.0- y el coste de sustituir dichos materiales por los del circuito USB, se ha realizado una comparación. De esta forma se podrá apreciar el ahorro que esto supondría:

Coste materiales programación			
Skymega1.0 y 2.0		Circuito USB	
Pasivos	-	Resistencias	0,42 €
Conectores ICSP y Puerto serie	2,5 €	Diodos	0,513 €
Cable + FTDI	6 €	Conector	1,07 €
Programador AVR	29,10 €	Pulsador	0,127 €
Total	37,6 €	Total	2,13 €

Tabla 6. Comparación de costes de material de programación

Como se puede observar, la diferencia de precios entre usar cable FTDI más programador AVR y usar el circuito USB, es bastante grande.

Esta diferencia no es del todo real, ya que no todo el que vaya a usar estas placas necesita el programador AVR (éste solo se usa una vez y puede ser prestado).

Por lo tanto, si no se tiene en cuenta el precio del programador AVR, la diferencia de precios sería aproximadamente de 5 €, una diferencia no muy grande.

Pruebas hardware

En un principio se probó a usar esta opción de programación y se montó el circuito en una placa protoboard, realizando una serie de pruebas hardware.

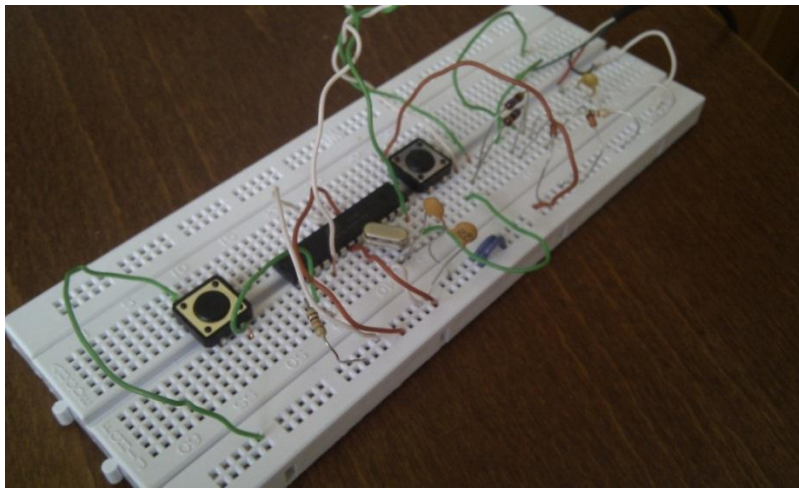


Imagen 72. Prueba hardware circuito USB

Debido a su complejidad, es necesario un estudio más profundo y detallado de cómo funciona este circuito, se decidió no seguir adelante y dejarlo como trabajo futuro.

Además, la diferencia de precios si no se tiene en cuenta el programador AVR, no es muy significativa.

De todas formas, se propone por incluir este circuito en la placa Skymega2.0 como trabajo futuro, ya que se trata de un trabajo muy amplio y no ha dado tiempo a implementarlo en este proyecto final de carrera.

3.4.2. Conector de un módulo XBee

La segunda mejora propuesta es la conexión de un módulo XBee. Esto se aumentará considerablemente el tamaño de la placa, por ello se propone colocarlo en un zócalo elevado, sobre otros componentes de perfil bajo, para así evitar que el tamaño aumente.

3.4.3. Posibilidad de conexión de una LiPo de más celdas

Una mejora sencilla pero muy útil, sería cambiar las resistencias que acondicionan la tensión de entrada del circuito de low voltage, y sustituirlas por dos potenciómetros, de tal forma que para cualquier tensión de entrada –dentro de los rangos de los que hemos hablado- siempre podamos tener una tensión de 2,7V a la entrada del detector de low voltage.

Esto nos permitiría colocar una batería LiPo de mayor tensión –mayor número de celdas-, controlando la descarga de ésta, y evitando que su tensión sobrepase la tensión mínima de descarga, es decir, evitando una descarga profunda¹².

¹² **Descarga profunda:** descarga muy por debajo del mínimo nivel soportado que causa el acortamiento de la vida útil de la batería LiPo.

Capítulo 4

Manual KiCad

4.1 Introducción

Como ya se ha indicado, todos los esquemáticos de diseño y el rutado de las placas, se han realizado con el software libre KiCad.

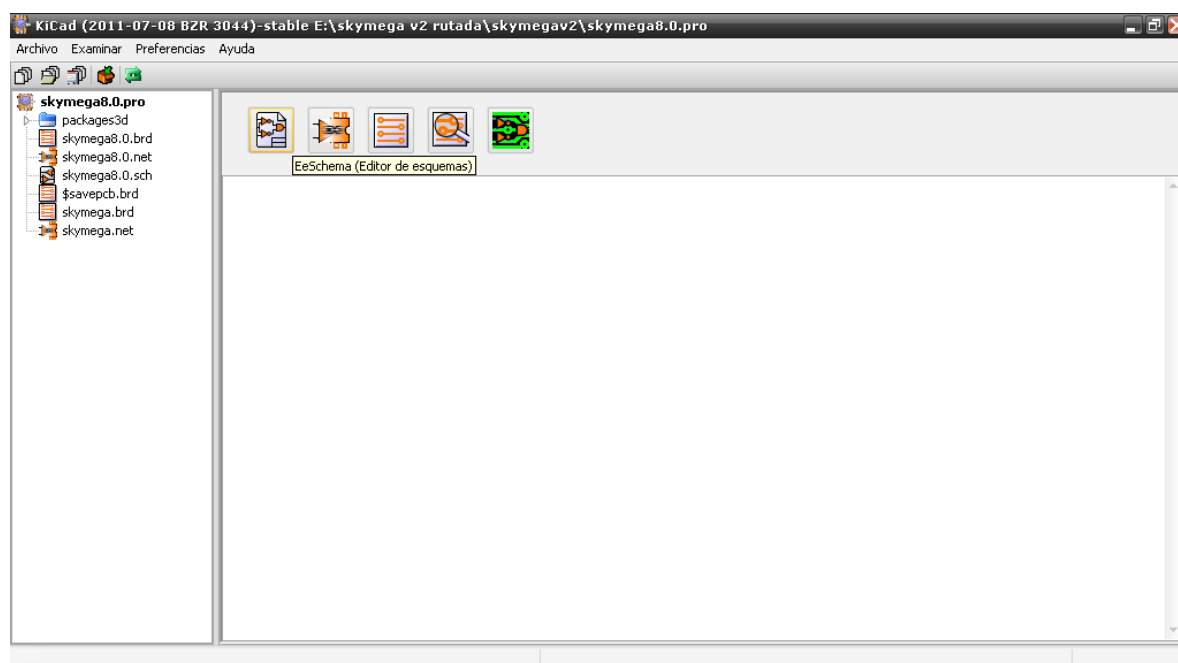
Debido a que ciertas partes del diseño y el rutado tienen una complejidad añadida, se ha decidido realizar un pequeño manual que facilite estas labores.

Sobre todo está pensado para aquellas personas que se inician en el diseño y rutado de pequeñas placas.

4.2 Diseño del esquemático

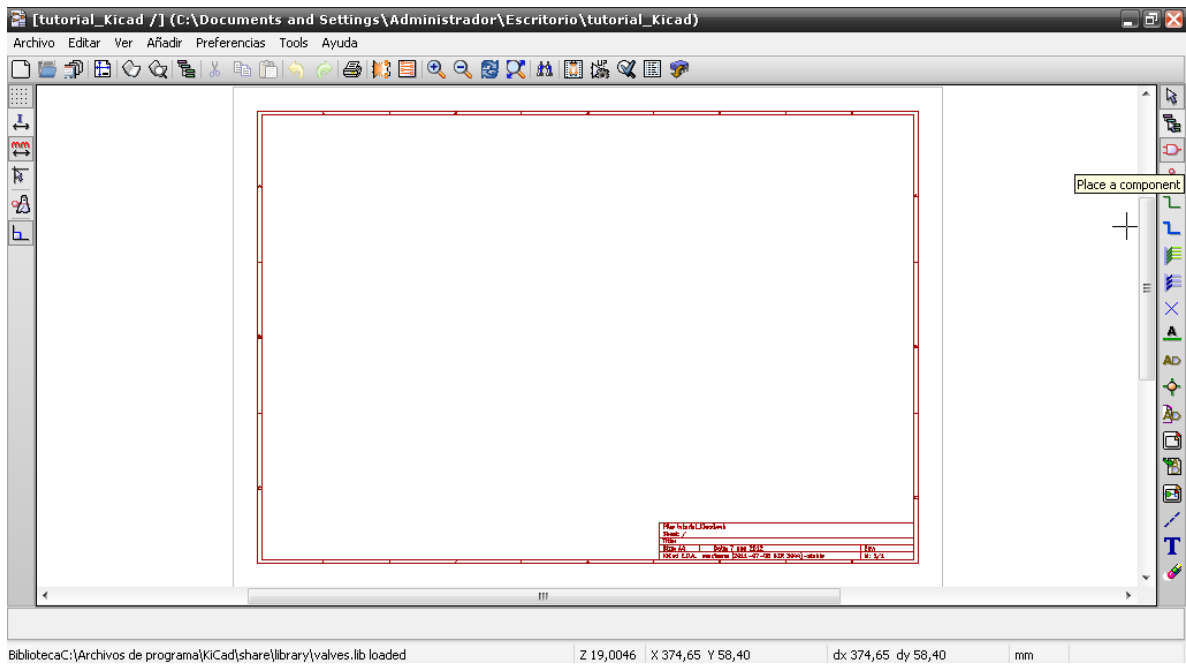
En primer lugar se deberá crear un nuevo proyecto “Crear un nuevo proyecto (.pro)”

Posteriormente se deberá abrir el editor de esquemas Eeschema¹³.

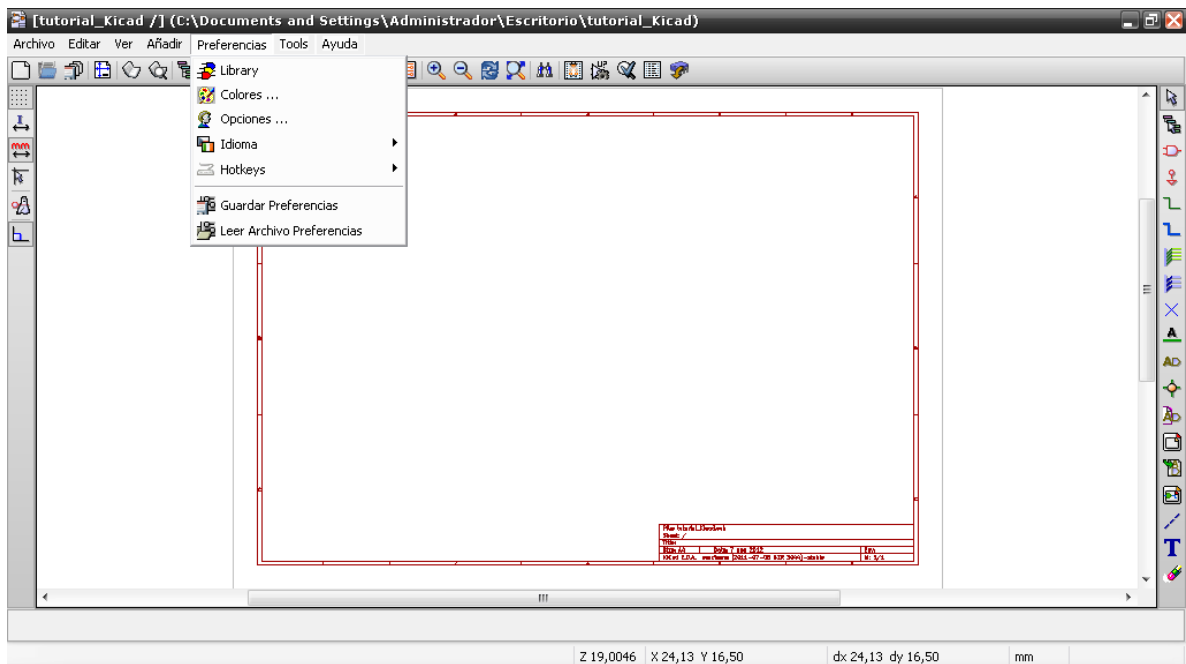


A continuación se irán colocando todos los componentes pinchando en el icono de la barra de la derecha “Place a component”, y buscándolos en las librerías.

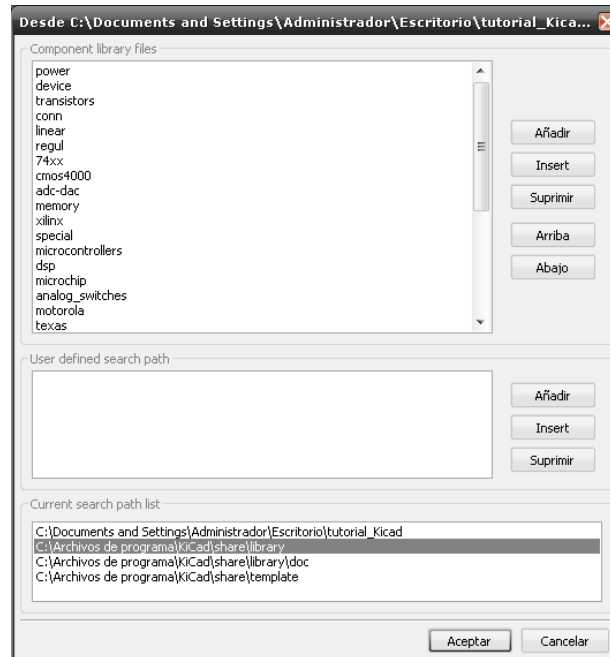
¹³ Eeschema es el editor de esquemas del programa KiCad. Mediante este editor se ha realizado el esquemático de la placa.



En el caso de que se quiera añadir una biblioteca en la barra superior se abre la pestaña "Preferencias" y "Library".



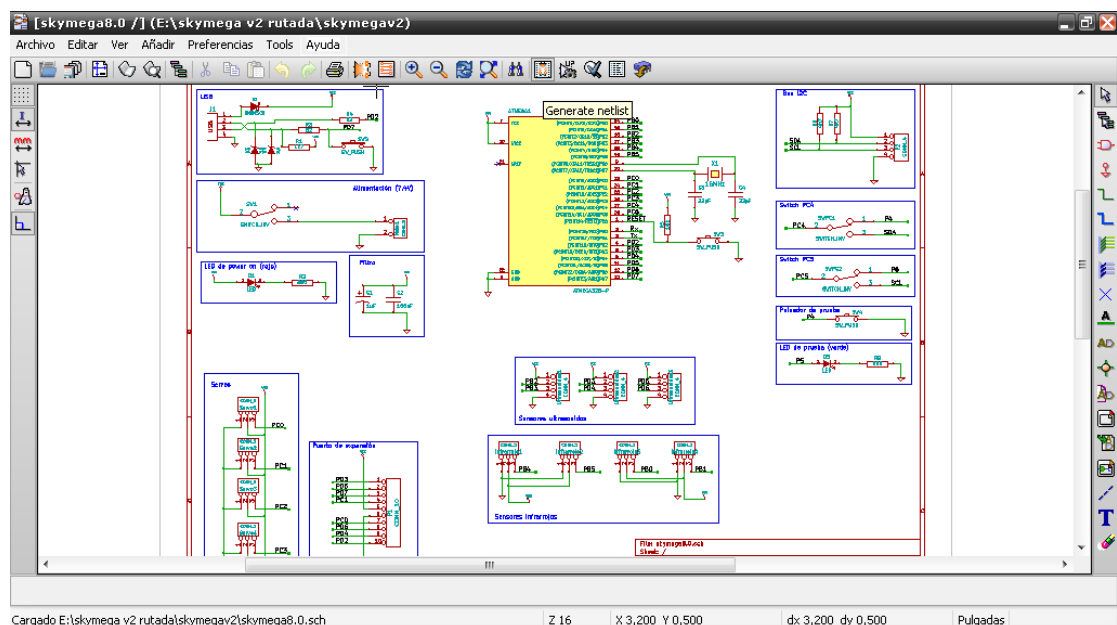
Se deberá pinchar en "añadir" y se buscará la carpeta donde está situada la biblioteca.



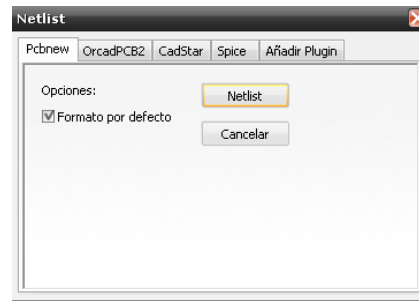
Una vez colocados todos los componentes, se deberá entrar en el icono "Annotate Schematic" mediante el cual se numerarán todos los componentes del esquemático.

4.3 Generación de la Netlist

Cuando ya están los componentes numerados se deberá guardar el esquemático y posteriormente darle a "Generate Netlist"

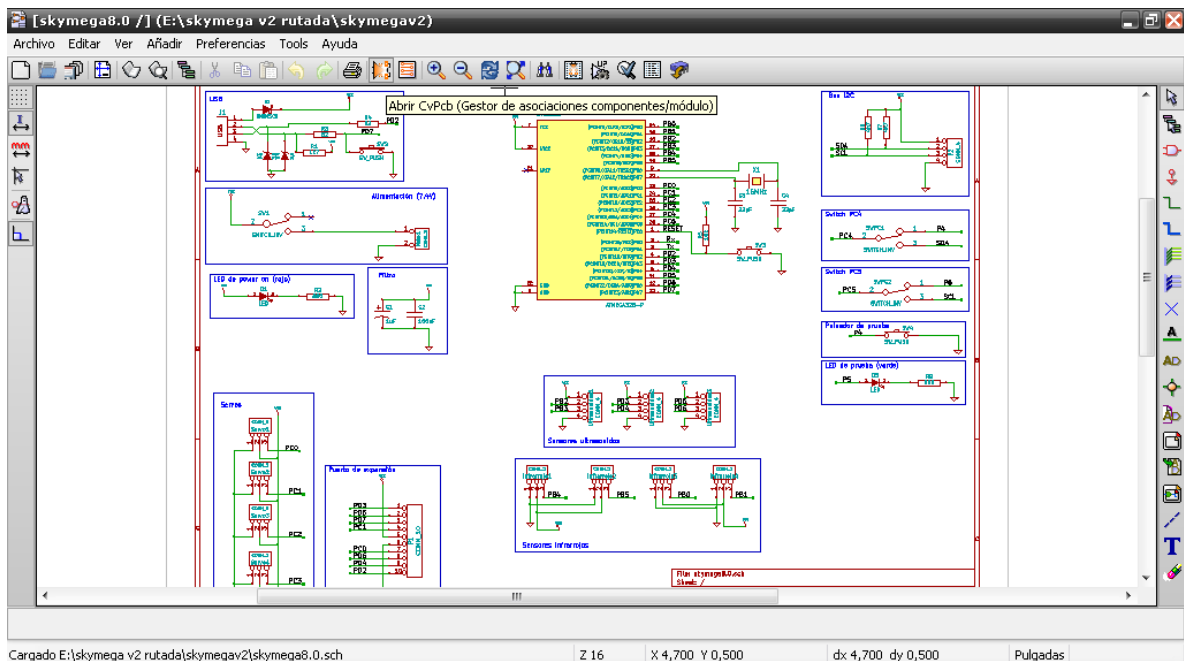


Aparecerá una pantalla en la que se deberá dar a "Netlist"

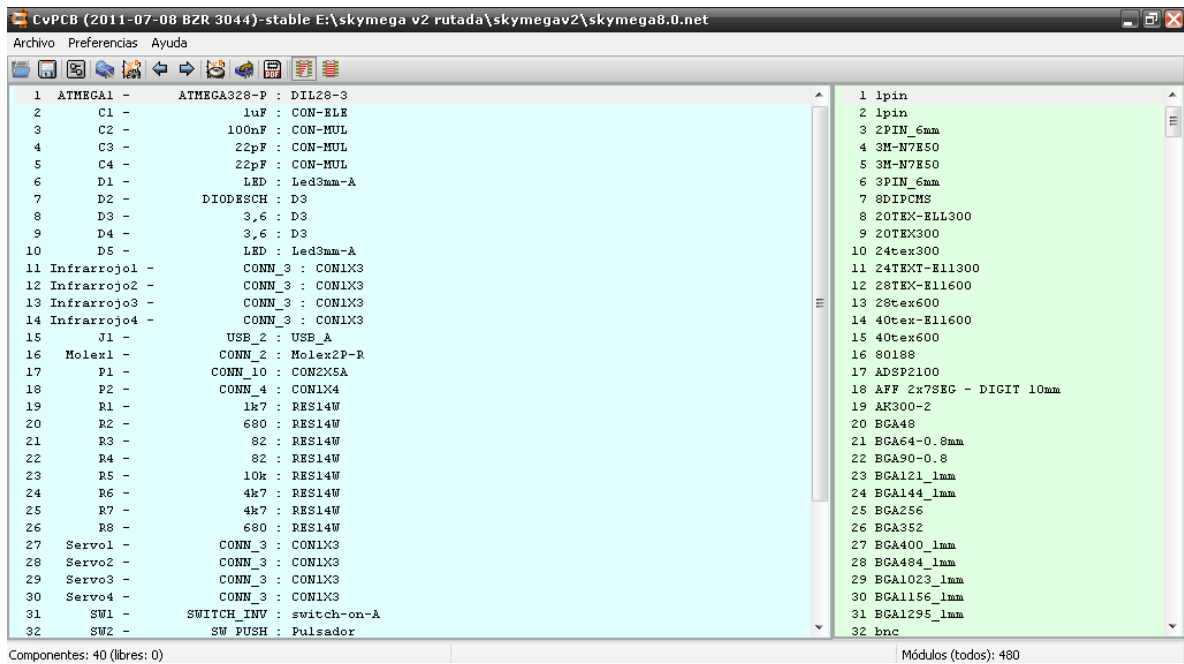


Una vez realizado esto, se guardará el archivo .net con el nombre y en el lugar deseados.

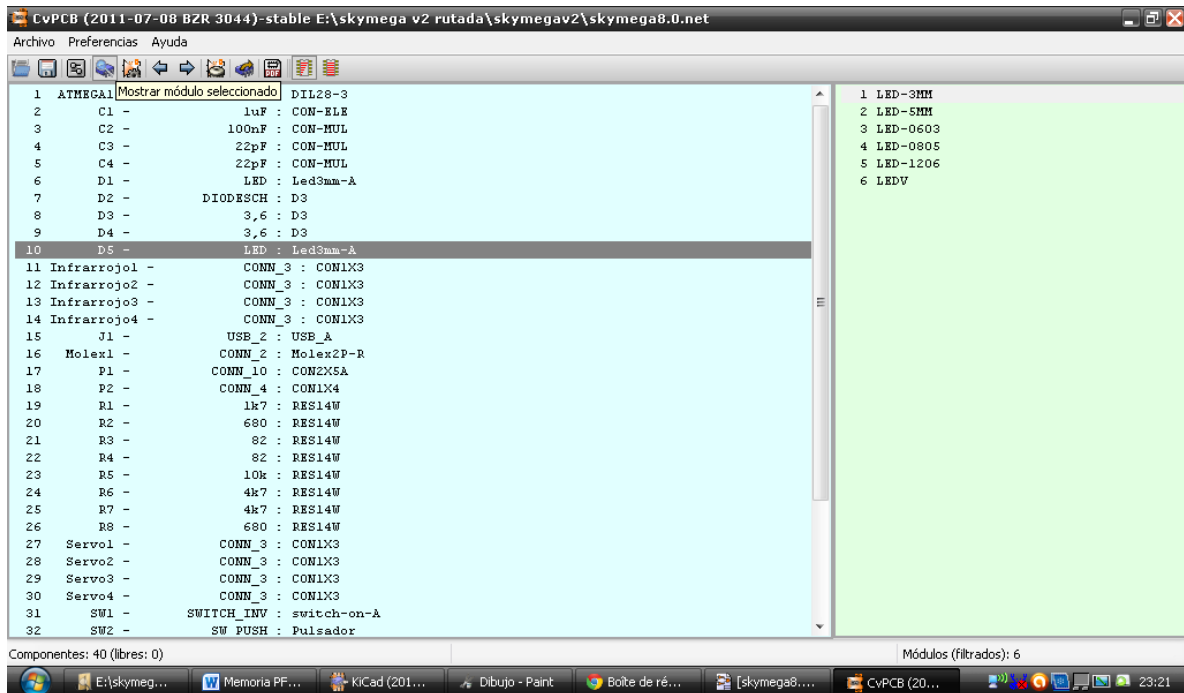
Una vez guardado el netlist se deberá dar en el icono en la barra superior "CvPCB (Gestor de asociaciones componentes/módulos)"



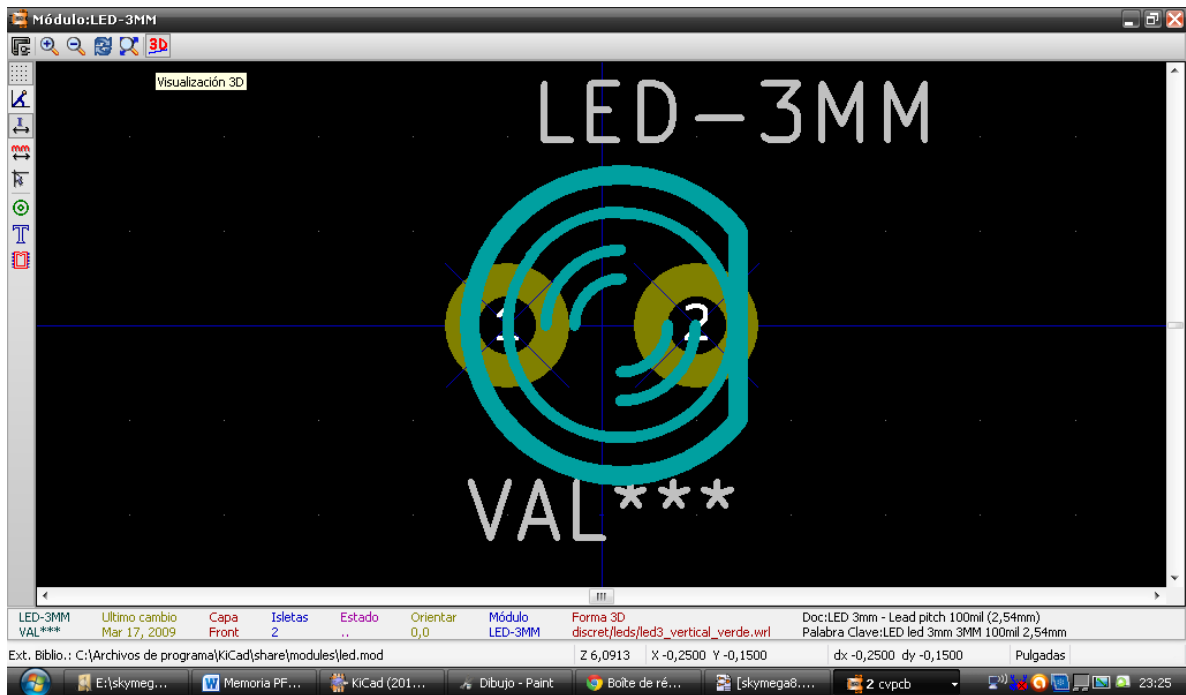
Se abrirá entonces este gestor y en él se deberá asociar a cada componente una huella de las que aparecen en el listado de la derecha.



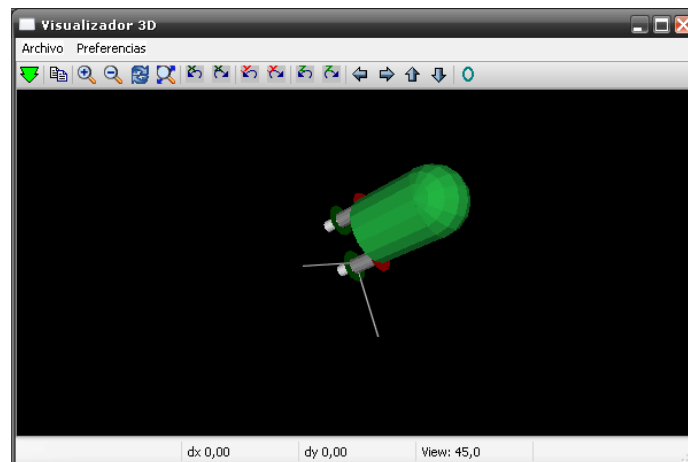
Se puede visualizar el módulo marcando el módulo, pinchando sobre él y pinchando en la barra superior sobre el icono "Mostrar módulo seleccionado".



Aparecerá entonces la siguiente pantalla, en la que en la barra superior se puede picar sobre el icono "Visualización 3D".



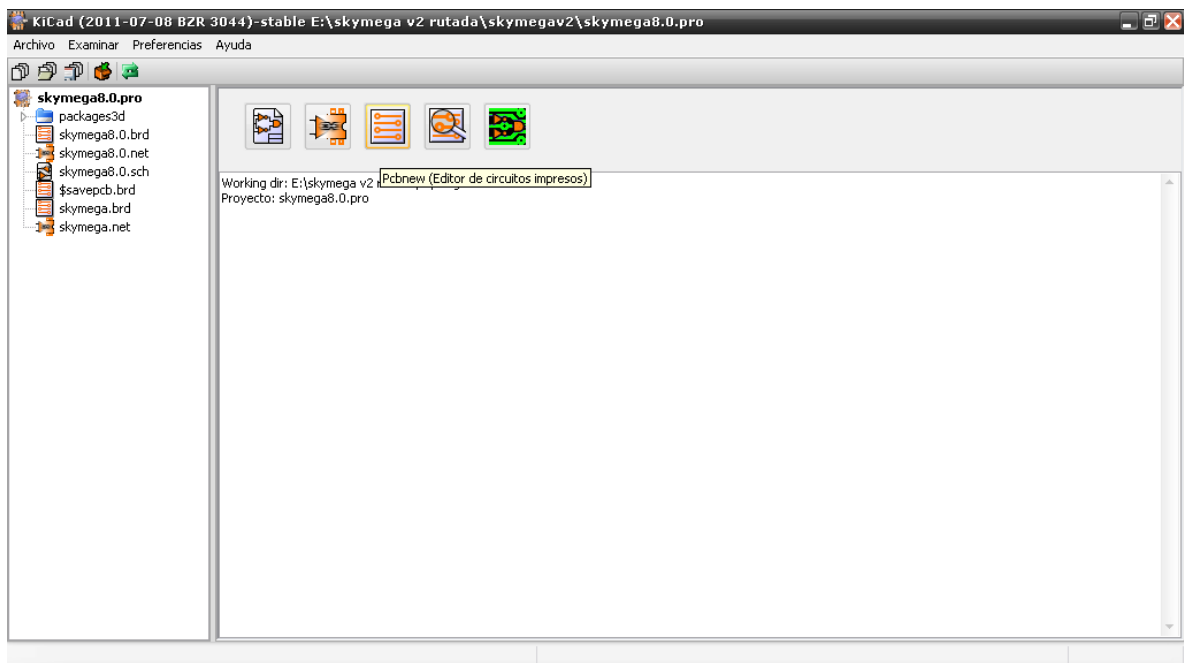
Aparecerá entonces la visualización en 3D del componente.



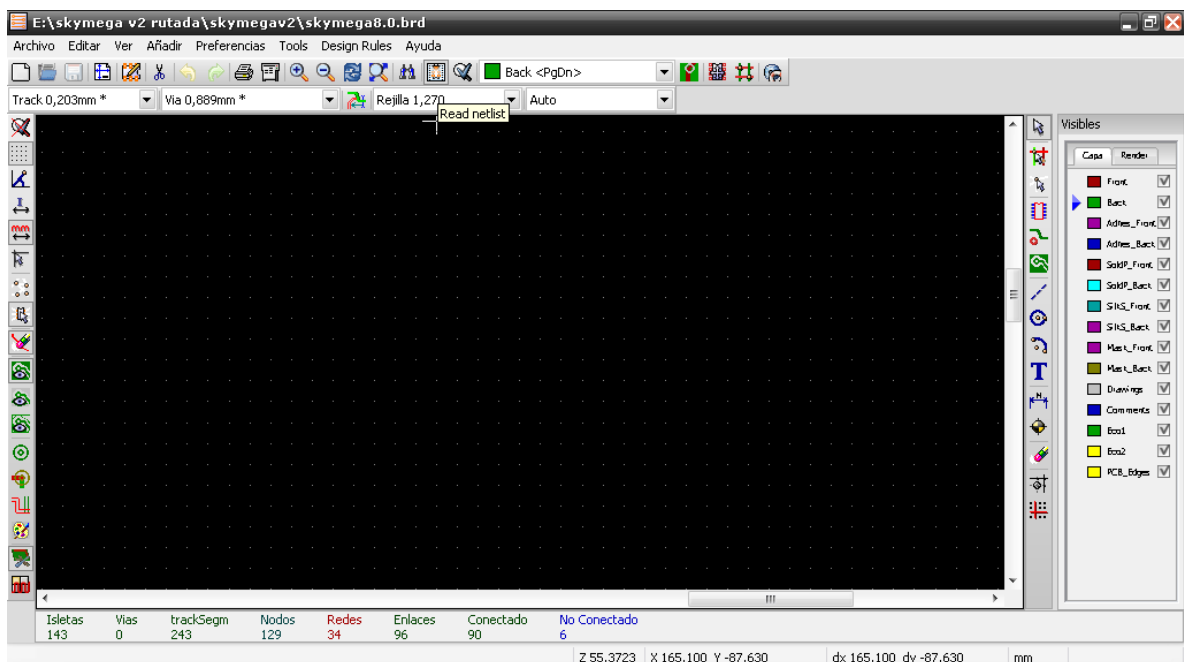
Una vez asociados los componentes se guardarán la netlist y las huellas y se cerrará la pantalla automáticamente.

4.4 Circuito impreso

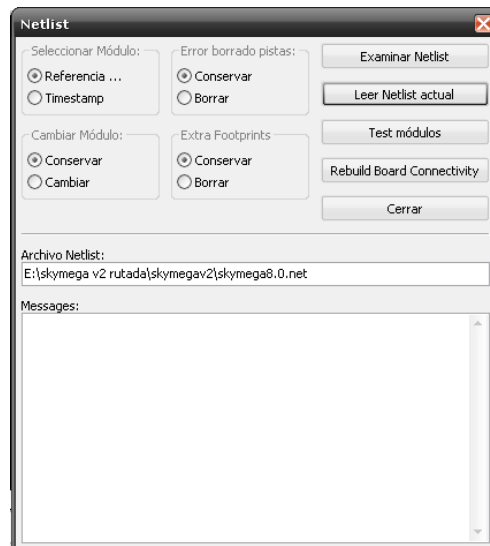
Para realizar el circuito impreso se deberá entrar en el icono Pcbnew (Editor de circuitos impresos).



Se abrirá entonces la pantalla siguiente y se deberá pulsar sobre el icono "ReadNetlist" de la barra de herramientas superior.



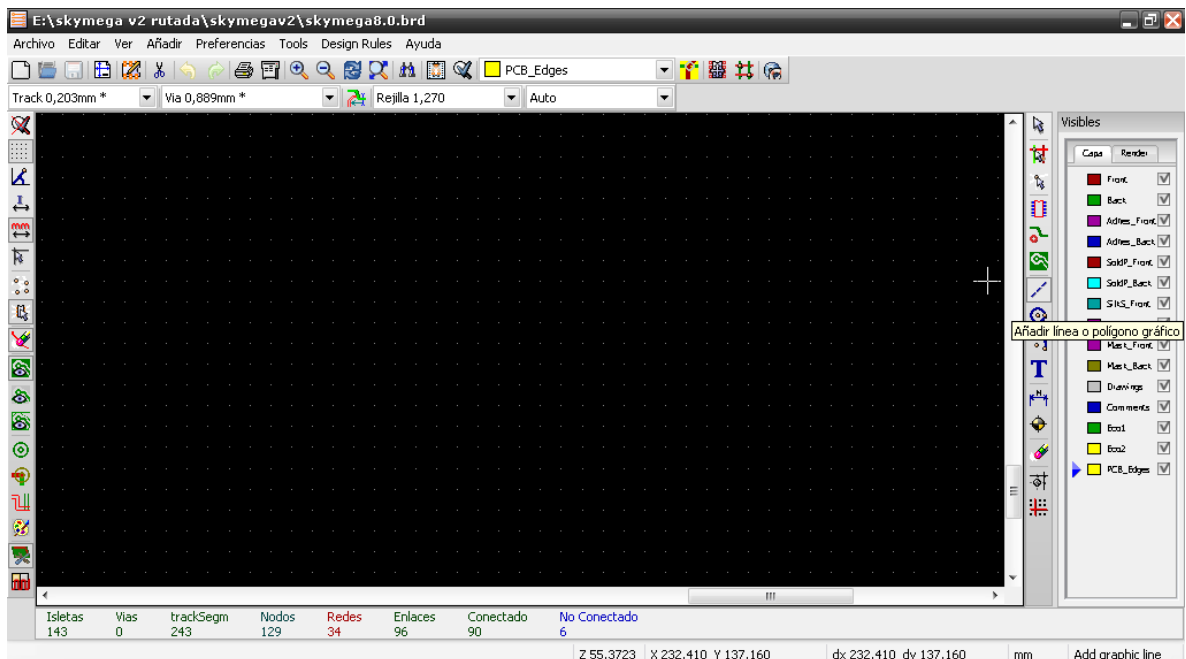
Aparecerá la pantalla siguiente en la que se deberá pulsar sobre "Examinar Netlist", "Leer Netlist" y "Test módulos".



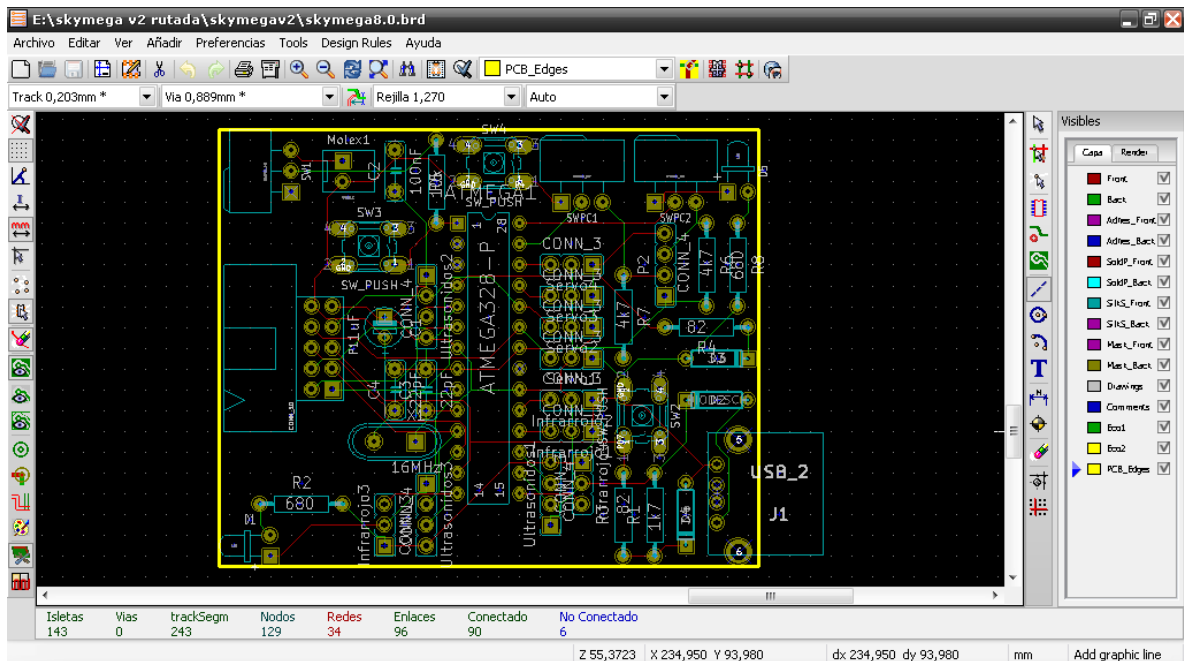
Una vez que se comprueba que no hay errores se deberá pulsar “Cerrar”.

Aparecerán todos los componentes entonces en la página.

Lo que se va a realizar a continuación será crear el borde de la placa. Para ello hay que situarse en la capa PCB_Edges. Posteriormente se deberá seleccionar la herramienta “Añadir línea o polígono” y dibujar los bordes de la placa.



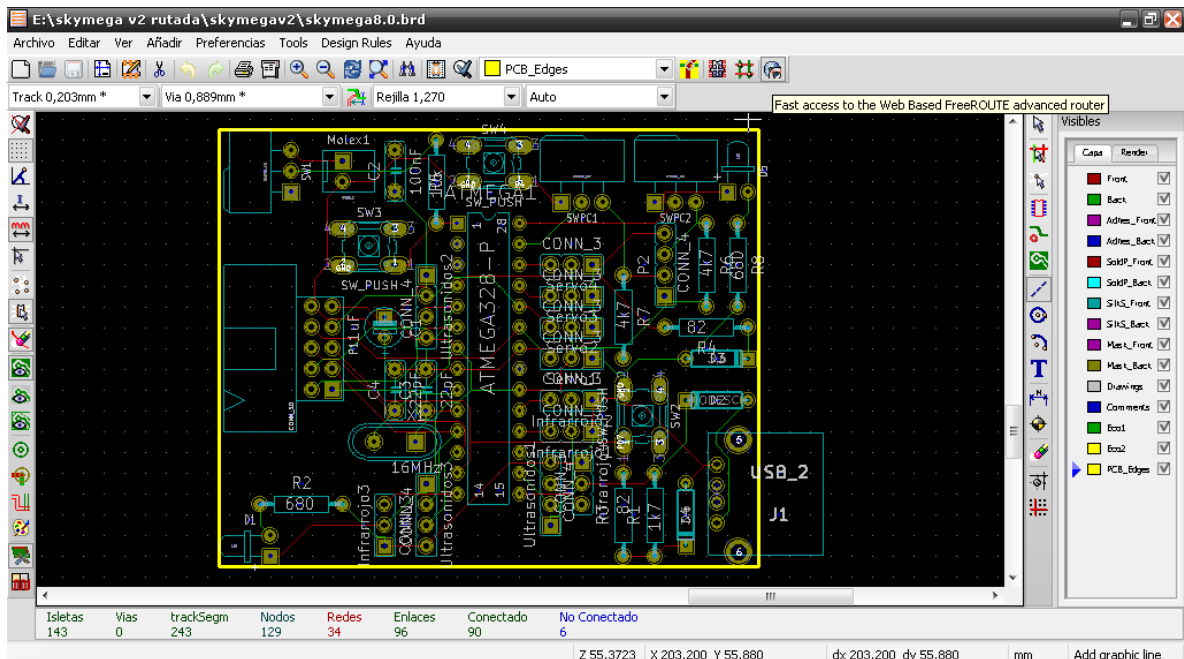
Una vez dibujados los bordes de la placa, se deberán colocar los componentes sobre ella. Para ello habrá que situarse en la placa de cobre “Front” y se moverán pinchando sobre cada componente y pulsando la letra “M”. Para rotar se pulsará la letra “R”.



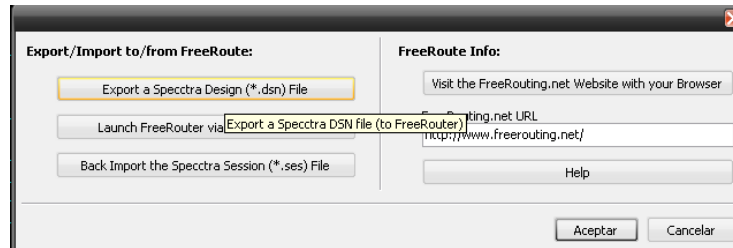
Una vez colocados los componentes sobre la placa se deberá proceder al rutado.

4.5 Rutado

Se ha realizado el auto ruteo con el programa Specctra. Para ello se pulsará sobre el icono "Fast Access to the Web Based Free ROUTE advanced router" de la barra de herramientas superior.



Aparecerá entonces la siguiente página y se deberá pulsar sobre "Export a Specctra Design (*.dsn) File."



Se guardará el archivo con extensión (.dsn) con el nombre y en el lugar deseados.

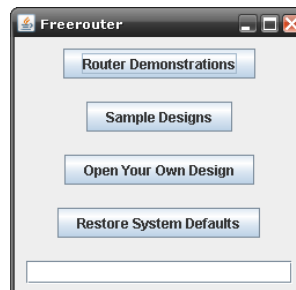
Una vez guardado se deberá pinchar en la misma ventana sobre "Visit the FreeRouting.net website with your Browser."

Se abrirá entonces la siguiente pantalla en el navegador.



Se deberá pinchar donde pone "Click here" y se descargará un archivo. (Nota: Importante tener descargado Java, si no se tiene descargado se puede descargar desde la misma página donde pone "You can download Java 6 from here".)

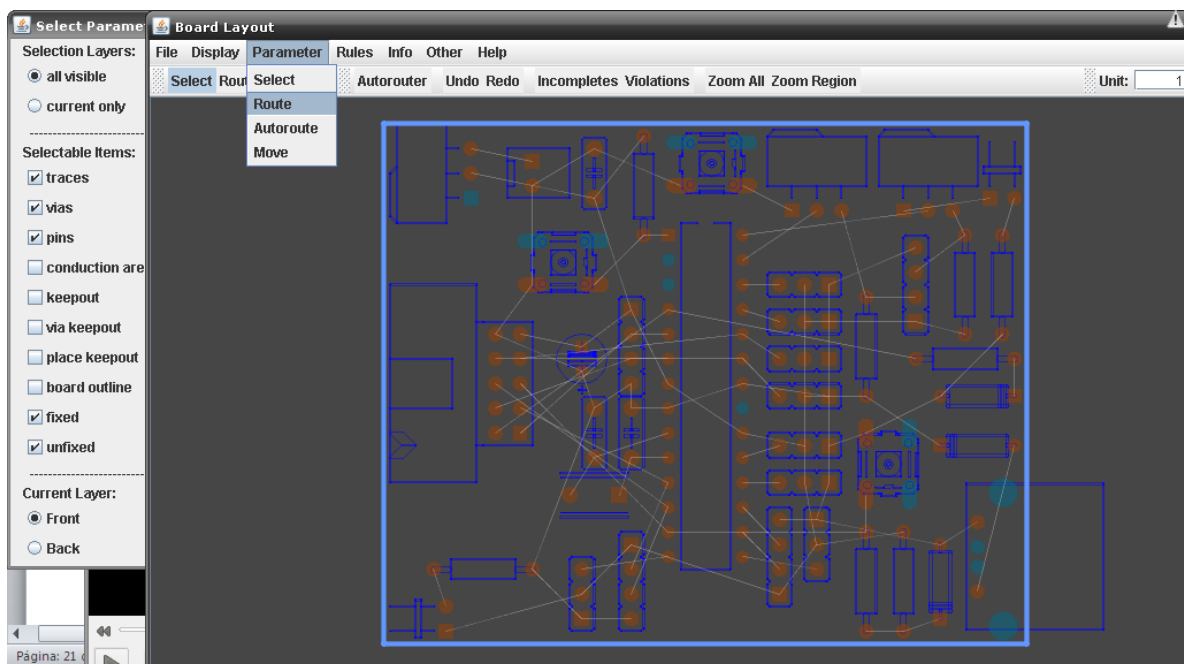
Se deberá abrir la aplicación descargada y aparecerá la siguiente ventana.



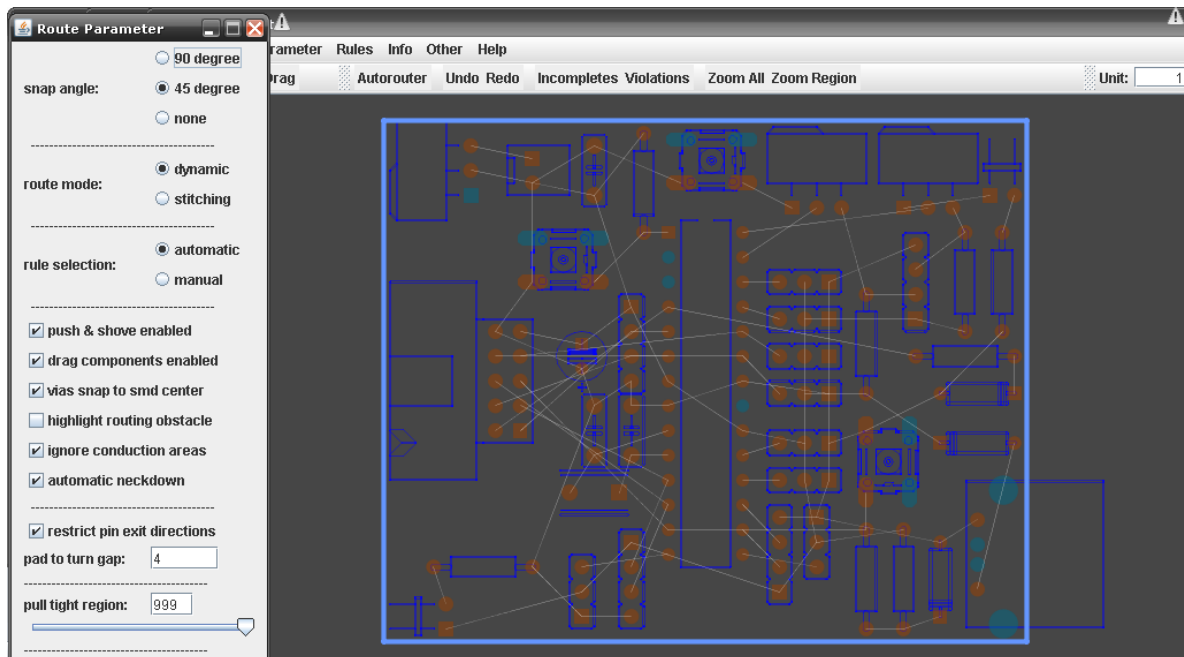
Una vez realizado esto se pulsará sobre "Open Your Own Design (open a board design in the Specctra .dsn format on your own local disc)"

Se deberá seleccionar entonces el archivo que se había guardado antes (.dsn)

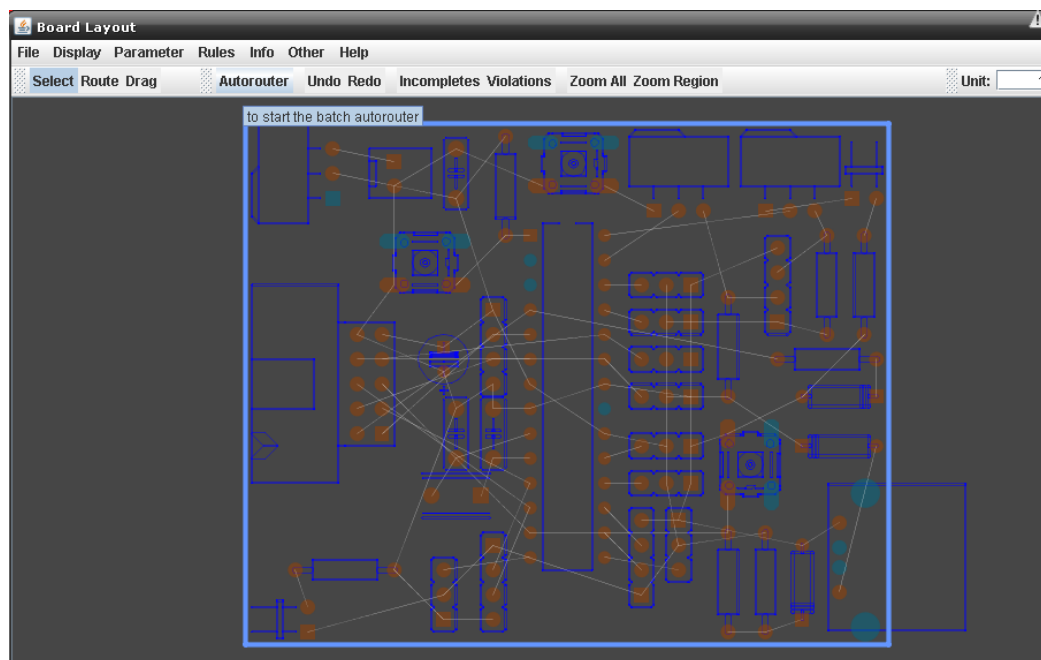
Posteriormente se abrirá la siguiente ventana y se deberá pinchar sobre "Parameter"-> "Route"



Aparecerá entonces la siguiente ventana en la que se deberán seleccionar los parámetros deseados. En este caso se seleccionaron los siguientes parámetros.

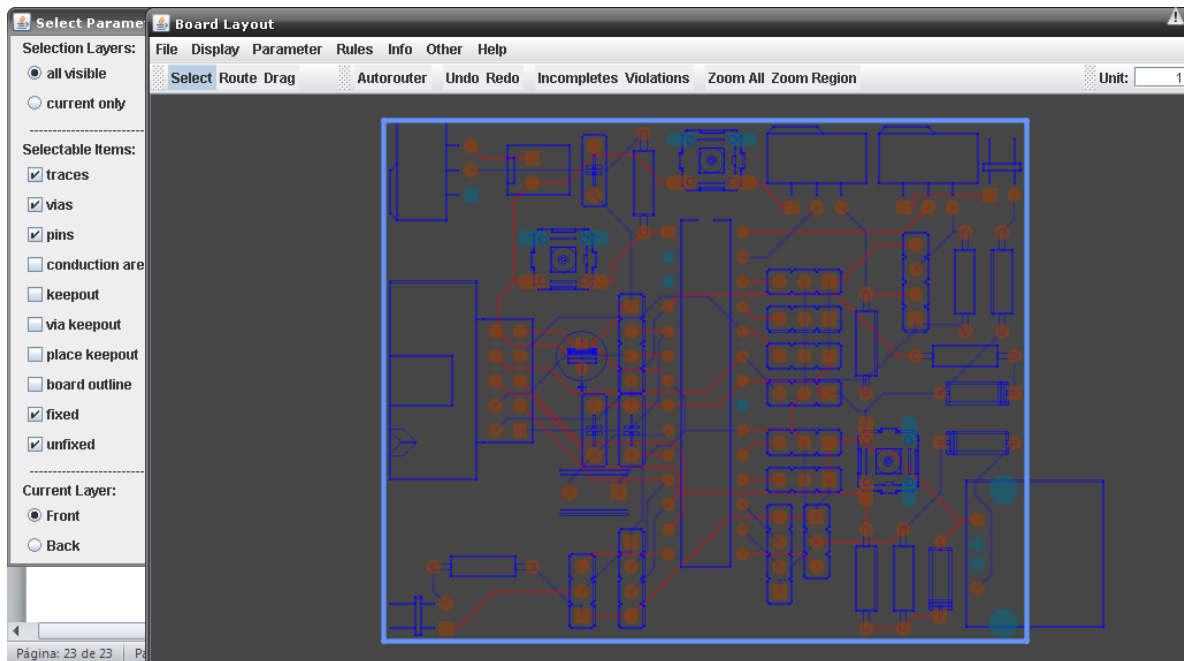


A continuación se pulsará sobre “Autorouter”

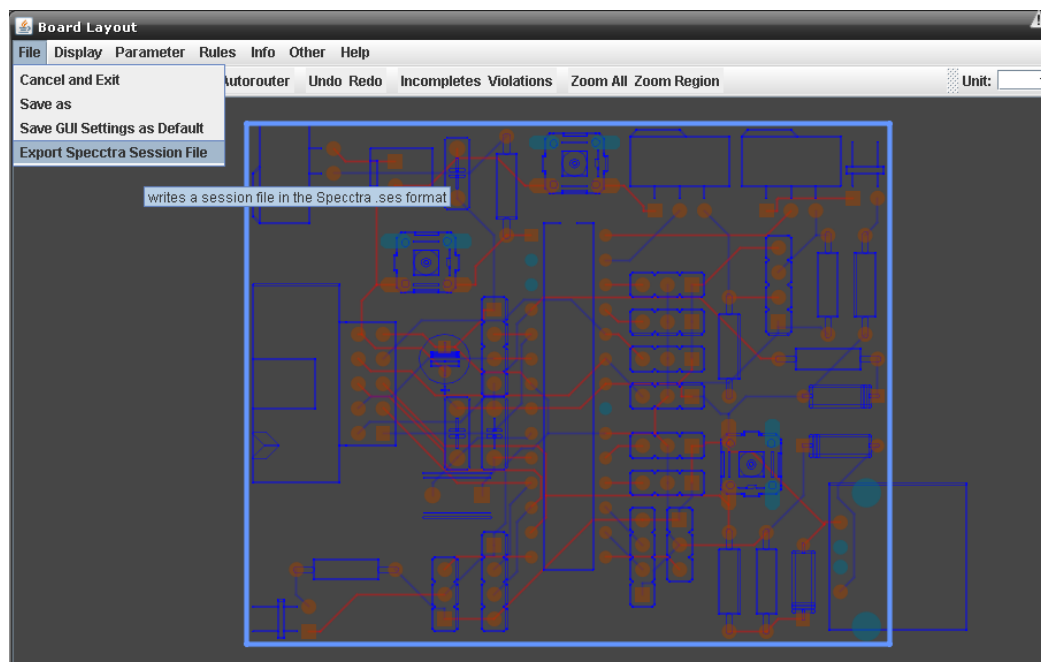


Entonces empezará el programa a realizar el ruteo automático. Se deberá esperar a que termine.

Una vez finalizado el ruteo en este caso aparecerán pistas de dos colores, ya que están en dos capas diferentes.

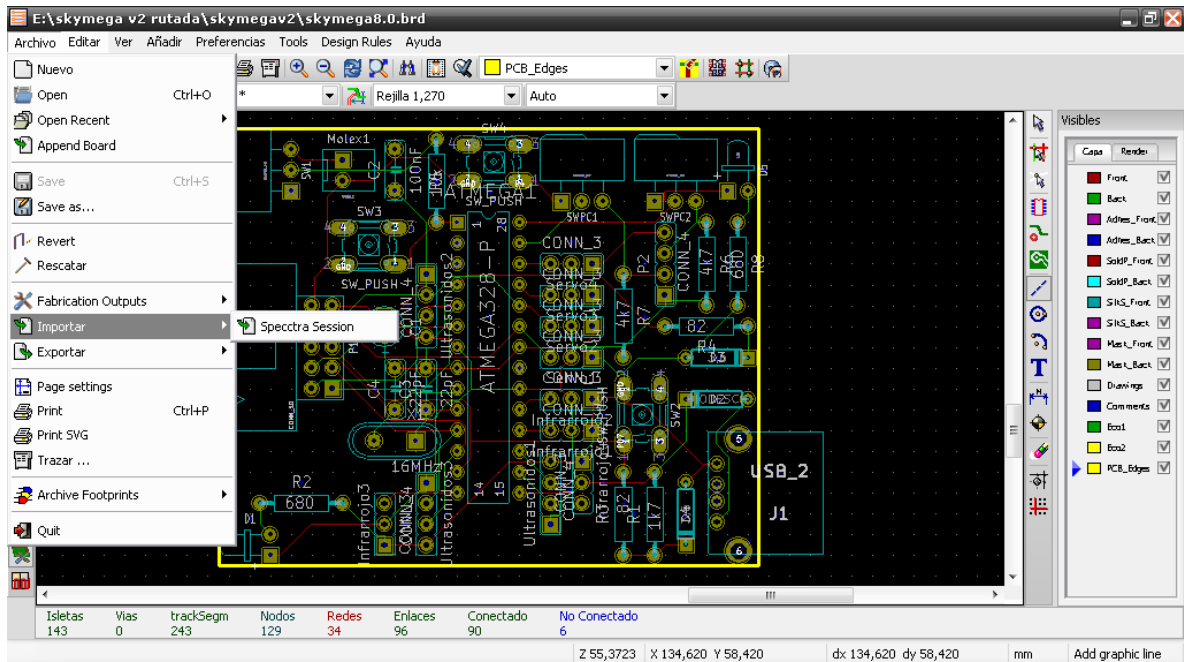


Una vez realizado el ruteo se exportará el archivo para utilizarlo en KiCad. Se deberá pinchar sobre "File" -> "Export Especctra Session File"

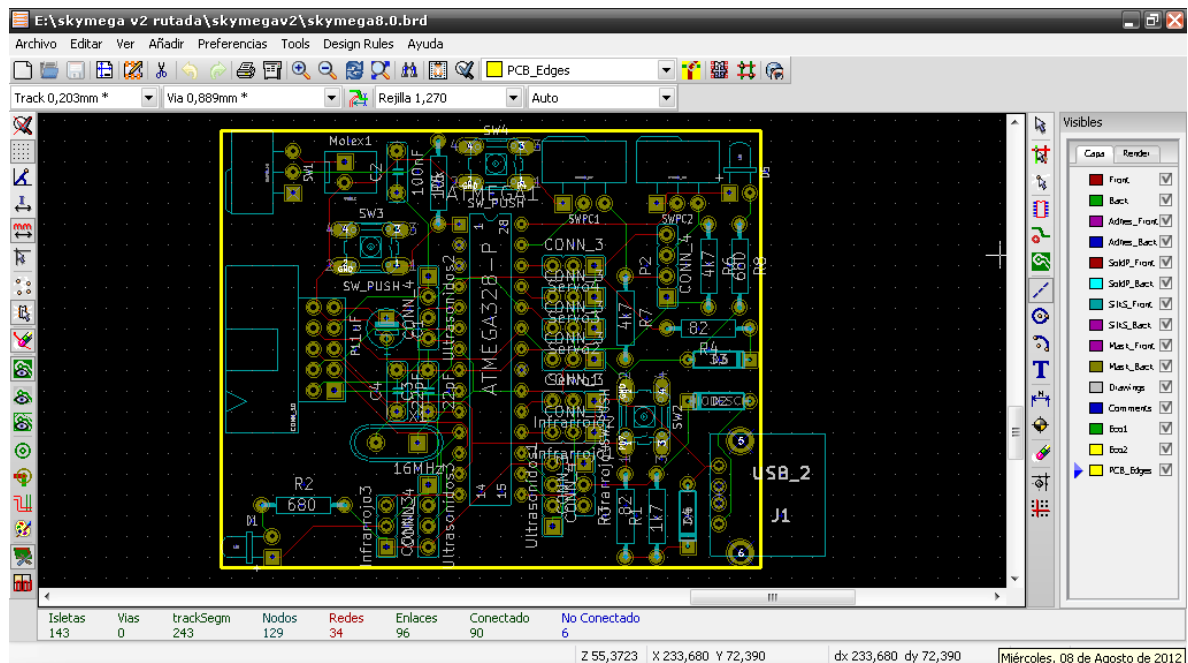


Se guardará el archivo con el mismo nombre que se le ha dado para exportarlo del KiCad y se le pondrá como extensión .ses.

Una vez guardado el archivo se debe volver al programa KiCad, al editor de circuitos impresos. Y pinchamos en “Archivo” de la barra de herramientas superior, después se debe pulsar sobre “Importar” -> “Specctra Session”. Se deberá abrir el archivo guardado en el Specctra.

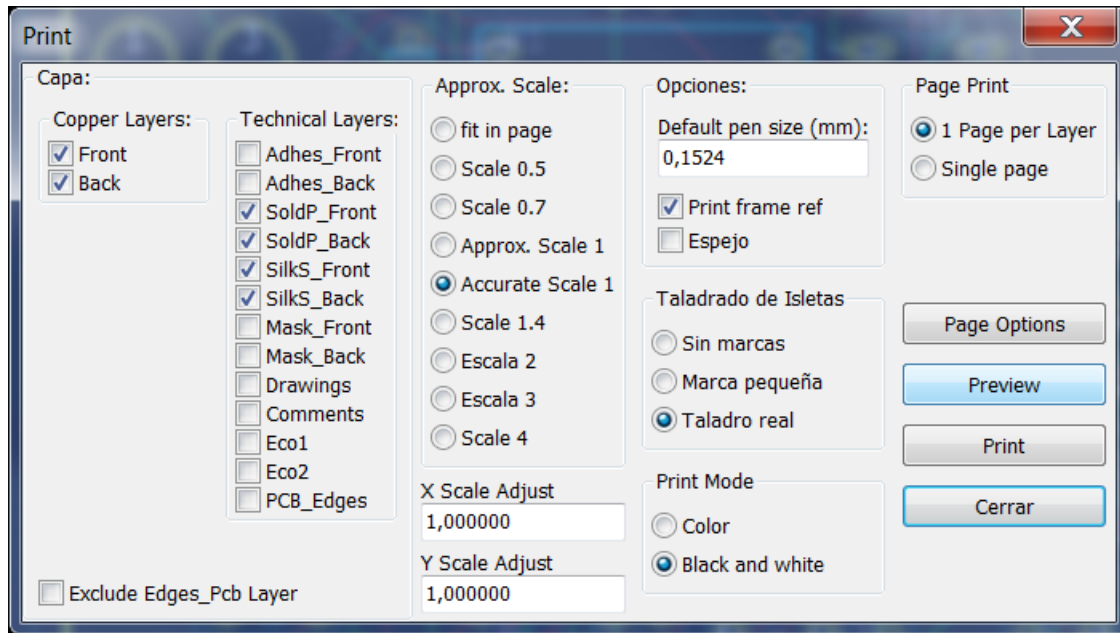


Aparecerá entonces en el KiCad la placa rutada, como se muestra en la siguiente figura.



4.6 Imprimir las capas de la placa

Una vez se tiene la placa rutada, para imprimir las capas de la placa se debe pulsar en el menú Archivo de la barra de herramientas superior, y posteriormente sobre "Print". Se abrirá entonces la siguiente ventana:



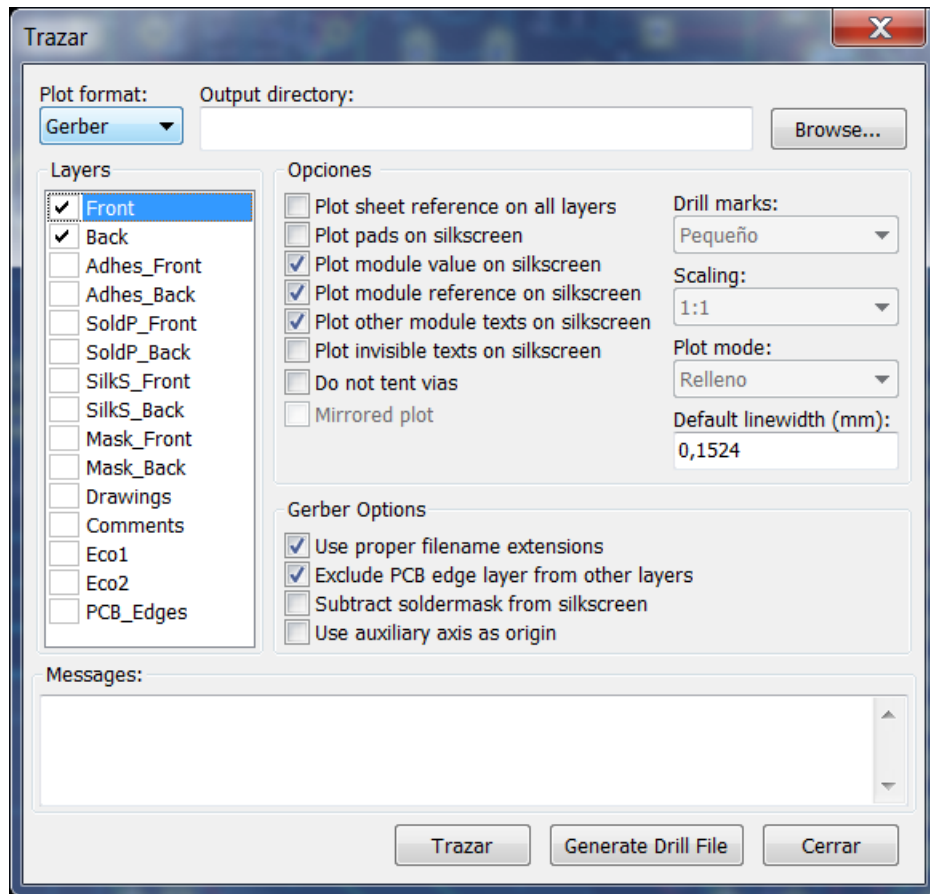
En esta ventana se deberán seleccionar los parámetros que se deseen. Se podrá previsualizar pulsando sobre la opción "Preview".

Una vez que ya se han seleccionado los parámetros deseados y se ha comprobado que es lo que se quiere imprimir, se deberá pulsar sobre "Print".

4.7 Generar los archivos Gerber

Para fabricar la placa es necesario generar unos archivos, que serán los que posteriormente se entregarán a la fábrica o al técnico del laboratorio para que realicen la placa.

Estos archivos se generarán pulsando sobre el menú Archivo de la barra de herramientas superior, y posteriormente sobre "Trazar". Se abrirá entonces la siguiente ventana:



En ella se deberá seleccionar el tipo de formato que se quiere, en este caso se trata de "Gerber". Se deberá también seleccionar el archivo donde se desea que se guarden los archivos generados, pinchando sobre "Browse".

Además de esto, se deberán elegir las capas deseadas-en nuestro caso Front y Back- y las opciones que se deseen.

Por último se deberá pulsar sobre "Generate Drill File" para generar el archivo de taladros.

Una vez realizado esto, se deberá seleccionar "Trazar" y se generarán de forma automática los archivos.

Capítulo 5

Estudio de costes

5.1 Introducción

Para el estudio de costes y de cara al presupuesto, los precios que se indican son orientativos. Se han tomado los precios del distribuidor RS Amidata¹⁴, ya que es un distribuidor profesional.

El criterio de búsqueda de los materiales ha sido mínimo coste unitario para el menor número de componentes, ya que este distribuidor obliga en ciertos casos a comprar más de un componente. No obstante, se sabe que se pueden encontrar los componentes por un precio menor.

Además, para abaratar el coste de la placa, en el caso de que se quieran realizar varias, se pueden hacer pedidos grandes de componentes, reduciéndose así el precio unitario de cada placa.

Otro aspecto a tener en cuenta es el coste de mano de obra del técnico del laboratorio al fabricar la placa. Este coste será elevado ya que se está realizando un prototipo, y todo el coste recae sobre éste. Cuando se realice la placa a nivel industrial, este coste se repartirá entre el número de placas fabricadas reduciéndose de manera significativa.

5.2 Comparativa general entre placas Skymega

En la siguiente tabla se muestran los costes de los componentes de las dos placas:

	Precio componentes
Skymega1.0	14,42
Skymega2.0	35,224

Tabla 7. Comparación de precios entre Skymega1.0 y Skymega2.0

Se han creado las dos placas de tal forma que el usuario puede elegir cuál de las dos adquirir según las funcionalidades que requiera, bien sea para un robot u otra aplicación.

La placa Skymega1.0, como se puede observar en la tabla mostrada anteriormente, es más económica que la skymega2.0, ya que esta última es más completa, y tiene un mayor número de componentes y funcionalidades. A pesar de su mayor coste, sigue estando muy por debajo de las placas existentes en el mercado, algo muy positivo y que cumple con el propósito buscado de diseñar una placa cuyo coste fuera de unos 30 €.

De esta forma, si necesita un robot que tenga las funcionalidades de la Skymega1.0 podría adquirir esta placa, gastándose así menos dinero.

¹⁴ Página web del distribuidor RS Amidata: es.rs-online.com/web/

Si, en cambio, se necesita un robot con un mayor número de funcionalidades se recurrirá a la Skymega2.0.

Se tiene que tener en cuenta que estos precios no son comparables con los de Arduino ya que la Skymega2.0 tiene un mayor número de funcionalidades y permite la conexión de dos motores de continua. Además, la Skymega2.0 no se ha fabricado aún de manera industrial. En el momento que esto ocurra disminuirá el precio respecto al de un prototipo, que es el que se detalla en este proyecto.

Para cargar el código en la placa Skymega1.0 se utilizó inicialmente un cable con el chip FTDI integrado, pero más tarde se encontró otra opción de la que ya hemos hablado (cable USB + módulo FTDI) más económica.

5.3 Precio componentes programación

En la siguiente tabla se muestra el desglose de los precios de los materiales que se utilizan para la programación de las placas Skymega1.0 y 2.0.

Aunque las pruebas realizadas con la Skymega1.0 se han realizado con el cable con el chip FTDI integrado, para abaratar la placa se propone utilizar el cable USB y el módulo FTDI.

Materiales programación Skymega1.0 y 2.0	
Conector acodado 2x3 pines	2,25 €
1x4 pines macho rectos	0,243 €
Programador AVR	29,10 €
Cable USB y módulo FTDI	6 €
TOTAL	37,60 €

Tabla 8. Materiales de programación Skymega1.0 y 2.0

Como se puede observar, lo que encarece las placas es el programador AVR. Aunque necesario, **no es imprescindible** comprarlo, ya que sólo se utiliza una vez, cuando el microprocesador es virgen, para cargarle el bootloader. Sería posible comprar un programador entre varias personas, y por tanto el coste de éste se dividiría en partes, siendo mucho menor.

Glosario

GND: Tierra del circuito.

VCC: Alimentación del circuito.

LED (*Light-Emitting Diode*): diodo emisor de luz.

Bus I2C (*Inter-Integrated Circuit*): bus de comunicaciones serie que permite la comunicación entre circuitos.

SDA: línea de datos del bus I2C.

SCL: línea reloj del bus I2C.

PCB (*Printed Circuit Board*): Tarjeta de circuito impreso.

USB (*Universal Serial Bus*): conector que permite comunicar y proveer de alimentación eléctrica a ordenadores, periféricos y dispositivos electrónicos.

Shield: placa que se coloca encima de una placa controladora para añadir funcionalidades a ella.

Batería LiPo: Batería de polímero de litio.

ICSP (*In Circuit Serial Programming*): puerto que sirve para cargar el bootloader al microprocesador cuando éste es virgen.

Bootloader: también llamado gestor de arranque, es un programa sencillo que no tiene la totalidad de las funcionalidades de un sistema operativo, y que está diseñado exclusivamente para preparar todo lo que necesita el sistema operativo para funcionar.

Programador AVR: dispositivo hardware que se utiliza para cargarle a los microcontroladores AVR el bootloader.

Microcontroladores AVR: familia de microcontroladores RISC del fabricante estadounidense Atmel.

Microcontrolador: circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

FTDI (*Future Technology Devices International*): chip que tomó su nombre de la empresa que lo desarrolló. Este cable sirve para descargar firmware en las placas, así como comunicarse vía puerto serie con ella mediante el ordenador o cualquier otro dispositivo.

I/O (*Input/Output*): Entrada/salida.

IR (*infrared*): sensores infrarrojos.

TX (*transmission*): pin a través del cual se transmiten datos.

RX (*reception*): pin a través del cual se reciben datos.

Netlist: archivo del programa de diseño de circuitos electrónicos con ordenador en el que se relaciona cada componente con su huella.

Rutado: realización de las pistas de la placa- uniones entre los componentes de la placa.

Archivos Gerber: archivos que es necesario generar, una vez diseñada la placa, para la fabricación de esta.

Bibliografía

Placas controladoras

- Arduino similar boards.
URL: <http://www.arduino.cc/playground/Main/SimilarBoards>
[Consultado en Febrero 2012]
- DFRobotShop Rover v2.
URL: <http://www.robotshop.com/ProductInfo.aspx?pc=rb-rbo-33>
[Consultado en Febrero 2012]
- BotBoarduino. URL: <http://www.lynxmotion.com/c-153-botboarduino.aspx>
[Consultado en Febrero 2012]
- Arduclema. URL: http://www.ray-ie.com/webray_015.htm
URL: <http://arduino.cc/forum/index.php/topic,30942.0.html>
[Consultado en Febrero 2012]
- BlimpDuino.
URL:
<http://diydrones.com/profiles/blog/show?id=705844%3ABlogPost%3A44817>
[Consultado en Febrero 2012]

Placas shield

- Motor Control v1.1. URL: <http://arduino.cc/en/Main/ArduinoMotorShield>
[Consultado en Febrero 2012]
- Arduino Ethernet Shield.
URL: <http://arduino.cc/es/Main/ArduinoEthernetShield>
[Consultado en Febrero 2012]
- Arduino Relay Shield.
URL: <http://www.bricogeek.com/shop/338-arduino-relay-shield.html>
[Consultado en Febrero 2012]
- Arduino grblShield v3.
URL: <http://www.synthetos.com/wiki/index.php?title=Projects:grblShield>
[Consultado en Febrero 2012]

- Rugged Motor Driver Shield.
URL: http://www.ruggedcircuits.com/html/rugged_motor_driver.html
[Consultado en Febrero 2012]

Puentes en H

- Control de motores de CC Puente H. Autor: Eduardo J. Carletti.
URL: http://robots-argentina.com.ar/MotorCC_PuenteH.htm
[Consultado en Mayo 2012]
- Puente en H para motor DC.
URL: <http://www.angelfire.com/co2/piel/puenteh.html>
[Consultado en Mayo 2012]

Reguladores de tensión

- Ejemplos reguladores de tensión.
URL:
https://sites.google.com/site/labarduino/recursos/componentes/reguladores_tension
[Consultado en Mayo 2012]

Librerías KiCad

- Librerías KiCad. URL: <http://www.kicadlib.org/>
[Consultado en Febrero 2012]
- Librerías KiCad. URL: <http://library.oshec.org/>
[Consultado en Febrero 2012]
- Búsqueda de componentes en bibliotecas KiCad.
URL: http://per.launay.free.fr/kicad/kicad_php/composant.php
[Consultado en Febrero 2012]

Manuales KiCad

- KiCad parte 1 normas de uso del programa- Youtube.
URL: http://www.youtube.com/watch?v=YA_5sIVdVY0
[Consultado en Febrero 2012]
- KiCad parte 2 normas de uso del programa.
URL: <http://www.youtube.com/watch?v=-t8qToplAXY&feature=endscreen>
[Consultado en Febrero 2012]
- KiCad parte 3 normas de uso del programa- Youtube.

URL:

<http://www.youtube.com/watch?feature=endscreen&NR=1&v=SF4YLwNDQtY>

[Consultado en Febrero 2012]

- KiCad parte 4 normas de uso del programa- Youtube.
URL: <http://www.youtube.com/watch?v=P9GTQMe6YYc&feature=related>
[Consultado en Febrero 2012]
- Diseño de circuitos impresos con KiCad.
URL: <http://www.ucontrol.com.ar/forosmf/tutoriales-guias-y-cursos-en-ucontrol/disenio-de-circuitos-impresos-con-kicad/?PHPSESSID=qvg26o30ophi1ech702vucom01>
[Consultado en Febrero 2012]

ATMEL

- A Quickstart Tutorial for ATMEL AVR Microcontrollers.
URL: <http://imakeprojects.com/Projects/avr-tutorial/>
[Consultado en Febrero 2012]

Sensores ultrasonido

- Sensores de ultrasonido.
URL: <http://iteadstudio.com/produce/cheap-cheap-ultrasonic-ranging-modulehc-sr04/>
[Consultado en Mayo 2012]

AVR

- AVR. URL : <http://es.wikipedia.org/wiki/AVR>
[Consultado en Febrero 2012]

Arduino

- Página oficial Arduino. URL: <http://arduino.cc/es/Guide/HomePage>
[Consultado en Febrero 2012]
- Introducción a la placa Arduino.
URL: <http://arduino.cc/es/Reference/Board>
[Consultado en Febrero 2012]
- Comparativa Arduino vs el resto.
URL: <http://www.neoteo.com/comparativa-arduino-arduino-vs-el-resto-15399>

[Consultado en Febrero 2012]

USBtinyISP

- USBtinyISP. URL: <http://www.ladyada.net/make/usbtinyisp/parts.html>
[Consultado en Febrero 2012]

Circuito USB

- Snoobie. URL: <http://www.frank-zhao.com/usnoobie/index.php>
[Consultado en Junio2012]

USB

- Universal Serial Bus. URL: http://es.wikipedia.org/wiki/Universal_Serial_Bus
[Consultado en Junio2012]

Skypic

- Skypic iearobotics.
URL: <http://www.earobotics.com/proyectos/skypic/skypic.html>
[Consultado en Junio2012]

Skymega1.0

- Skymega Thingiverse.
URL: <http://www.thingiverse.com/thing:14197>
[Consultado en Junio2012]
- Skymega1.0 iearobotics.
URL: <http://www.earobotics.com/wiki/index.php?title=SkyMega>
[Consultado en Junio2012]

Anexos

Anexo 1. Presupuesto Skymega1.0

COD.	UNI.	Descripción	MED.	P. UNI.	P. TOTAL
01	CAP.	MATERIAL			
01.1.	SUBCAP.	Material Skymega1.0			
	Ud.	Condensador cerámico 22pF	2	0,263	0,526
	Ud.	Condensador cerámico 100nF	1	0,182	0,182
	Ud.	Condensador electrolítico 1uF/16v	1	0,182	0,182
	Ud.	Conector acodado 2x5 pines	1	2,51	2,51
	Ud.	Conector acodado 2x3 pines	1	2,25	2,25
	Ud.	LED verde de 3mm	1	0,12	0,12
	Ud.	LED rojo de 3mm	1	0,162	0,162
	Ud.	Conector Molex macho recto de 2 pines (2,54mm)	1	0,476	0,476
	Ud.	Tira de 40 pines macho rectos	1	2,43	2,43
	Ud.	Tira de 40 pines macho acodados	1	0,668	0,668
	Ud.	Resistencia de 1/4W 10k	1	0,014	0,014
	Ud.	Resistencia de 1/4W 680	2	0,014	0,028
	Ud.	Resistencia de 1/4W 4K7	2	0,019	0,038
	Ud.	Pulsador para circuito impreso	2	0,127	0,254
	Ud.	Interruptor de 2 posiciones acodado para CI	1	0,77	0,77
	Ud.	Cristal de cuarzo 16MHz	1	0,206	0,206
	Ud.	Zócalo de 2x14 pines (300 mils)	1	0,754	0,754

	Ud.	Microcontrolador Atmega168 de ATMEL	1	2,85	2,85
		Total subcapítulo 1			14,42
01.2.	SUBCAP.	Material Complementario			
		Programador AVR	1	29,10	29,10
		Pila AAA	4	0,318	1,27
		Conector Molex hembra recto de 2 pines (2,54mm)	1	0,24	0,24
		Cable con chip FTDI integrado	1	15,20	15,20
		Total subcapítulo 2			45,81
		TOTAL CAPÍTULO 01			60,23 €
02.	CAP.	Fabricación			
02.01.01		Precio de fabricación de la placa			3,95€
		TOTAL CAPÍTULO 02			3,95€
		PRECIO TOTAL de la placa			78,6€

Anexo 2. Presupuesto Skymega2.0

COD.	UNI.	Descripción	MED	P. UNI. €	P. TOTAL
01	CAP.	MATERIAL			
01.1.	SUBCAP.	Material Skymega2.0			
	Ud.	Condensador cerámico 22pF	2	0,263	0,526
	Ud.	Condensador cerámico 100nF	4	0,182	0,728
	Ud.	Condensador electrolítico 1uF/16v	1	0,182	0,182
	Ud.	Condensador cerámico/multicapa 100UF	1	0,228	0,228
	Ud.	Conector acodado 2x5 pines	1	2,51	2,51
	Ud.	LED verde de 3mm	1	0,12	0,12
	Ud.	LED rojo de 3mm	2	0,162	0,324
	Ud.	Conector Molex macho recto de 2 pines (2,54mm)	2	0,476	0,952
	Ud.	Conector acodado 2x3 pines	1	2,25	2,25
	Ud.	Integrado L293 (puentes en H)	1	3,08	3,08
	Ud.	Conector Jack Hembra	1	2,50	2,50
	Ud.	Tira de 40 pines macho rectos	1	2,43	2,43
	Ud.	Tira de 40 pines macho acodados	1	0,668	0,668
	Ud.	Resistencia de 1/4W 10K	3	0,014	0,042
	Ud.	Resistencia de 1/4W 680	1	0,014	0,014
	Ud.	Resistencia de 1/4W 4k7	2	0,019	0,038
	Ud.	Resistencia de 1/4W 348	2	0,60	1,20
	Ud.	Resistencia de 1/4W 1k	1	0,027	0,027

Ud.	Resistencia de 1/4W 15k	1	0,038	0,038
Ud.	Diodo de polarización inversa 200mA	1	0,171	0,171
Ud.	Componente TC54	1	0,50	0,50
Ud.	Módulo PTH08000WAZT	1	8,71	8,71
Ud.	Pulsador para circuito impreso	2	0,127	0,254
Ud.	Interruptor de 2 posiciones acodado para CI	1	0,77	0,77
Ud.	Interruptor de 2 posiciones recto	2	1,02	2,04
Ud.	Cristal de cuarzo 16 MHz	1	0,206	0,206
Ud.	Zócalo de 2x14 pines (300 mils)	1	0,754	0,754
Ud.	Microcontrolador Atmega328P de ATMEL	1	4,31	4,31
Total subcapítulo 1				35,22€

01.2. SUBCAP. Material Complementario

Programador AVR	1	29,10	29,10
Batería Lipo	1	6,77	6,77
Conector Jack Macho	1	0,86	0,86
Módulo FTDI con cable USB a micro USB	1	6	6
Total subcapítulo 2			42,73€

TOTAL CAPÍTULO 01 77,95€

02.	CAP.	Implementación	
02.01.01		Precio de fabricación de la placa PCB.	3,95€
TOTAL CAPÍTULO 02			3,95€
PRECIO TOTAL de la placa			81,90€

Anexo 3. Referencia componentes

Se adjunta como anexo una tabla en la que aparecen las referencias de los componentes de la tienda RS para facilitar la búsqueda en caso de quererse fabricar estas placas.

3.1. Referencias Skymega1.0

Descripción	MED.	Código fabricante
Componentes		
Condensador cerámico 22pF	2	736-8833
Condensador cerámico 100nF	1	653-0181
Condensador electrolítico 1uF/16v	1	704-7854
Conector acodado 2x5 pines	1	718-9462
Conector acodado 2x3 pines	1	718-9080
LED verde de 3mm	1	247-1375
LED rojo de 3mm	1	228-5001
Conector Molex macho recto de 2 pines (2,54mm)	1	483-8461
Tira de 40 pines macho rectos	1	718-9674
Tira de 40 pines macho acodados	1	156-033
Resistencia de 1/4W 10k	1	707-7745
Resistencia de 1/4W 680	2	707-7984
Resistencia de 1/4W 4K7	2	707-7745
Pulsador para circuito impreso	2	758-1957
Interruptor de 2 posiciones acodado para CI	1	712-2558
Cristal de cuarzo 16MHz	1	693-6970
Zócalo de 2x14 pines (300 mils)	1	813-187

Microcontrolador Atmega168 de ATMEL	1	715-3952
Material Complementario		
Programador AVR	1	696-2563
Pila AAA	4	221-793
Conector Molex hembra recto de 2 pines (2,54mm)	1	296-4934
Cable con chip FTDI	1	687-7786

3.2. Referencias Skymega2.0

Descripción	MED	Código fabricante
Componentes		
Condensador cerámico 22pF	2	736-8833
Condensador cerámico 100nF	4	653-0181
Condensador electrolítico 1uF/16v	1	704-7854
Condensador electrolítico 100UF	1	520-1393
Conector acodado 2x5 pines	1	718-9462
LED verde de 3mm	1	247-1375
LED rojo de 3mm	2	228-5001
Conector Molex macho recto de 2 pines (2,54mm)	2	483-8461
Conector acodado 2x3 pines	1	718-9080
Integrado L293 (puentes en H)	1	526-868
Conector Jack Hembra	1	487-836
Conector Jack Macho	1	505-1564

Tira de 40 pines macho rectos	1	718-9674
Tira de 40 pines macho acodados	1	156-033
Resistencia de 1/4W 10K	3	707-7745
Resistencia de 1/4W 680	1	707-7984
Resistencia de 1/4W 4k7	2	707-7745
Resistencia de 1/4W 348	2	135-825
Resistencia de 1/4W 1k	1	135-847
Resistencia de 1/4W 15k	1	135-932
InvPoldiode 200mA	1	508-270
Componente TC54	1	207-0051
Módulo PTH08000WAZT	1	620-1136
Pulsador para circuito impreso	2	758-1957
Interruptor de 2 posiciones acodado para CI	1	712-2558
Interruptor de 2 posiciones recto	2	712-2554
Cristal de cuarzo 16 MHz	1	693-6970
Zócalo de 2x14 pines (300 mils)	1	813-187
Microcontrolador Atmega328P de ATMEL	1	696-2260

Anexo 4. Hojas de características

4.1. TC54

<http://docs-europe.electrocomponents.com/webdocs/0027/0900766b800279fe.pdf>

4.2. Módulo PTH08000WAZT

<http://docs-europe.electrocomponents.com/webdocs/082b/0900766b8082b356.pdf>

4.3. Integrado L293

<http://docs-europe.electrocomponents.com/webdocs/0b76/0900766b80b7662c.pdf>

4.4. Microcontrolador Atmega328/168

<http://docs-europe.electrocomponents.com/webdocs/0db4/0900766b80db4deb.pdf>